## Contents

How to Use This Manual .......................................................... ix
How the Manual is Organized .................................................. x
Conventions .............................................................................. xi
Related Manuals ...................................................................... xii

1 Introduction ........................................................................ 1-1
   Report Format ...................................................................... 1-2
      Heading Line .................................................................. 1-2
      Report Body .................................................................. 1-2
      Message Line .................................................................. 1-3
   Components of Ultimate RECALL ........................................ 1-5

2 Forming an Ultimate RECALL Statement ............................... 2-1
   Relational Operators .......................................................... 2-4
      LIKE Operator ................................................................ 2-7
   Wild Cards ........................................................................ 2-9
   Files ................................................................................. 2-11
   Itemlists ........................................................................... 2-12
      Explicit Itemlist ............................................................ 2-12
      Implicit Itemlist ............................................................ 2-12
   Select-Lists ..................................................................... 2-13
   Selection Criteria ............................................................. 2-15
   Sort Criteria ..................................................................... 2-18
   Multivalued Attributes ......................................................... 2-20
   BY-EXP and Select Lists ..................................................... 2-22
   Output Specifications .......................................................... 2-23
      Controlling and Dependent Attributes ......................... 2-24
   Print Limiters .................................................................... 2-26
   Modifiers .......................................................................... 2-28
      BREAK-ON .................................................................... 2-30
      COL-HDR-SUPP ............................................................ 2-33
      DBL-SPC ....................................................................... 2-33
      DET-SUPP ..................................................................... 2-34
      FOOTING ....................................................................... 2-35
      GRAND-TOTAL ............................................................... 2-37
      HDR-SUPP ..................................................................... 2-38
      HEADING ...................................................................... 2-39
      ID-SUPP ........................................................................ 2-42
Contents

LPTR................................................................. 2-42
NOPAGE......................................................... 2-42
ONLY.............................................................. 2-43
SUPP.............................................................. 2-44
TAPE.............................................................. 2-44
TOTAL............................................................ 2-45
USING............................................................ 2-47
WITHIN......................................................... 2-48
Throwaway Modifiers........................................ 2-50

Options........................................................... 2-51

3 Ultimate RECALL Commands........................... 3-1
CHECK-SUM.................................................. 3-4
COUNT.......................................................... 3-6
HASH-TEST................................................... 3-7
ISTAT............................................................. 3-9
LIST.............................................................. 3-11
LIST-ITEM...................................................... 3-14
LIST-LABEL.................................................... 3-17
REFORMAT...................................................... 3-21
  Reformatting to Tape.................................... 3-23
S-DUMP.......................................................... 3-25
SELECT.......................................................... 3-28
SORT............................................................. 3-30
SORT-ITEM..................................................... 3-34
SORT-LABEL................................................... 3-37
SREFORMAT..................................................... 3-41
  Reformatting to Tape.................................... 3-43
SSELECT....................................................... 3-45
STAT.............................................................. 3-47
SUM.............................................................. 3-49
T-DUMP.......................................................... 3-51
T-LOAD.......................................................... 3-54

4 Attribute Definition Items............................... 4-1
Components of Attribute Definition Items.............. 4-3
  Item-ID......................................................... 4-3
  D/CODE - Definition Code............................. 4-5
  A/AMC - Attribute Number............................. 4-5
  S/NAME - Column Heading.............................. 4-5
  S/AMC - Structure Code................................. 4-6
  V/CONV - Conversion Code............................. 4-7
  V/CORR - Correlative Code............................. 4-7
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>V/TYP - Justification</td>
<td>4-8</td>
</tr>
<tr>
<td>V/MAX - Column Width</td>
<td>4-8</td>
</tr>
<tr>
<td>Effects of V/TYP on Sorting</td>
<td>4-10</td>
</tr>
<tr>
<td><strong>5 Processing Codes</strong></td>
<td></td>
</tr>
<tr>
<td>Correlatives and Conversions</td>
<td>5-1</td>
</tr>
<tr>
<td>Multiple Processing Codes</td>
<td>5-5</td>
</tr>
<tr>
<td>A - Arithmetic Code</td>
<td>5-6</td>
</tr>
<tr>
<td>Order of Operations</td>
<td>5-9</td>
</tr>
<tr>
<td>Totals</td>
<td>5-10</td>
</tr>
<tr>
<td>Averages</td>
<td>5-11</td>
</tr>
<tr>
<td>B - BASIC Subroutine Call Code</td>
<td>5-13</td>
</tr>
<tr>
<td>C.DATA</td>
<td>5-14</td>
</tr>
<tr>
<td>Reserved BASIC Operations</td>
<td>5-14</td>
</tr>
<tr>
<td>C - Concatenation Code</td>
<td>5-18</td>
</tr>
<tr>
<td>D - Date Code</td>
<td>5-19</td>
</tr>
<tr>
<td>F - Function Code</td>
<td>5-22</td>
</tr>
<tr>
<td>F-Code Stack</td>
<td>5-25</td>
</tr>
<tr>
<td>Totals</td>
<td>5-28</td>
</tr>
<tr>
<td>Averages</td>
<td>5-29</td>
</tr>
<tr>
<td>G - Group Code</td>
<td>5-31</td>
</tr>
<tr>
<td>L - Length Code</td>
<td>5-32</td>
</tr>
<tr>
<td>MC - Mask Character Code</td>
<td>5-33</td>
</tr>
<tr>
<td>MD - Mask Decimal Code</td>
<td>5-35</td>
</tr>
<tr>
<td>ML - Mask Decimal, Left Justify Code</td>
<td>5-36</td>
</tr>
<tr>
<td>MP - Mask Packed Decimal Code</td>
<td>5-39</td>
</tr>
<tr>
<td>MR - Mask Decimal, Right Justify Code</td>
<td>5-41</td>
</tr>
<tr>
<td>MT - Mask Time Code</td>
<td>5-44</td>
</tr>
<tr>
<td>MX - Mask Hexadecimal Code</td>
<td>5-46</td>
</tr>
<tr>
<td>P - Pattern Match Code</td>
<td>5-47</td>
</tr>
<tr>
<td>R - Range Code</td>
<td>5-49</td>
</tr>
<tr>
<td>S - Substitution Code</td>
<td>5-50</td>
</tr>
<tr>
<td>T - Text Extraction Code</td>
<td>5-52</td>
</tr>
<tr>
<td>Tfile - Translation File Code</td>
<td>5-53</td>
</tr>
<tr>
<td>V - Item-ID Extraction Code</td>
<td>5-57</td>
</tr>
<tr>
<td>Using Ultimate RECALL with Report Forms</td>
<td>6-1</td>
</tr>
<tr>
<td>Features of Report Forms</td>
<td>6-3</td>
</tr>
<tr>
<td>Forms Expression</td>
<td>6-5</td>
</tr>
<tr>
<td>Print Specifications</td>
<td>6-8</td>
</tr>
<tr>
<td>Audit File</td>
<td>6-10</td>
</tr>
<tr>
<td>Windows</td>
<td>6-12</td>
</tr>
<tr>
<td>Modifiers in Forms Output</td>
<td>6-15</td>
</tr>
</tbody>
</table>
BREAK-ON ................................................................. 6-15
COL-HDR-SUPP ......................................................... 6-15
DBL-SPC ................................................................. 6-15
GRAND-TOTAL .......................................................... 6-15
HDR-SUPP ................................................................. 6-16
HEADING and FOOTING .............................................. 6-16
ID-SUPP ................................................................. 6-16
ONLY ................................................................. 6-16
TOTAL ................................................................. 6-16
Options in Forms Output .............................................. 6-17
A Option - Aligning a Form ......................................... 6-17
B Option - Defining a Background .................................. 6-18
M Option - Multiple Reports per Page ......................... 6-19
Z Option - Resetting Page Number ................................ 6-20
Placing Data ............................................................. 6-21
S/AMC ................................................................. 6-21
V/TYP ................................................................. 6-21
V/MAX ................................................................. 6-22
Null Values ............................................................... 6-24
Example of Form .......................................................... 6-26
Sequence of Prompts ..................................................... 6-30
Hints .............................................................................. 7-1
Performance ............................................................... 7-2
Printing Dependent Attributes ....................................... 7-5
Suppressing Display of Data .......................................... 7-8
Suppressing Data on Detail Lines ..................................... 7-8
Suppressing Data on Detail Lines and Break Lines ............. 7-9
Suppressing All Details .................................................. 7-10
BREAK-ON and Forms ................................................ 7-11
Testing for Existence .................................................. 7-12
Verifying Data ............................................................ 7-13
Glossary ....................................................................... A-1
Default Attribute Definition Items .................................. B-1
Database Used in Examples ........................................... C-1
ASSEMBLY File .......................................................... C-2
CUSTOMERS File ...................................................... C-3
INVOICE File ........................................................... C-6
PROD.NO File ........................................................... C-14
ZIP File ................................................................. C-18
Index

Figures
1-1.  Columnar Report Format .............................................. 1-3
1-2.  Non-Columnar Report Format ..................................... 1-4
4-1.  Multiple Attribute Definition Items .............................. 4-10
4-2.  Effects of V/TYP on Sorting ...................................... 4-11
5-1.  F-Code Stack Activity to Resolve F
      F;C3;C4;C20;+;* .................................................................. 5-26
5-2.  F-Code Stack Activity to Resolve
      F;2;3;*;C#;:;1;C3;C2;[]: .................................................................. 5-27
6-1.  Sample Forms Report ..................................................... 6-2
6-2.  DICT INVOICE Items ..................................................... 6-26
6-3.  PROC to Create Reports ............................................... 6-27
6-4.  Sample Multipaged Form ............................................... 6-28

Tables
1-1.  Connectives ................................................................. 1-6
1-2.  Ultimate RECALL Commands ........................................ 1-9
2-1.  Relational Operators .................................................... 2-5
2-2.  Wild Cards ................................................................. 2-9
2-3.  Modifiers ................................................................. 2-29
2-4.  Options ................................................................. 2-52
3-1.  Ultimate RECALL Commands ........................................ 3-2
4-1.  Attribute Definition Item Format .................................... 4-2
4-2.  Characters Not Recommended for Item-IDs .................... 4-4
4-3.  V/TYP Codes ............................................................. 4-9
5-1.  Processing Codes ......................................................... 5-2
5-2.  RCL.COMMON Variables ............................................. 5-15
5-3.  The C.FLAGS Variable ................................................... 5-15
6-1.  Effects of V/TYP and V/MAX on Forms ......................... 6-23
6-2.  Sequence of Prompts ................................................... 6-30
B-1.  Default Attribute Definition Items ............................. B-2
How to Use This Manual

This manual is a complete description of Ultimate RECALL. It is intended for anyone who uses the Ultimate System. If you wish to use Ultimate RECALL to produce an output report, chapters 2, 3 and 6 would be most useful in telling you how to do this. If you wish to set up a database, or understand the elements in your database, chapters 4 and 5 provide you with information on these topics.

The following applications also make use of Ultimate databases:

- UltiKit® application development environment.
- Ultimate UPDATE® screen-oriented, online database maintenance functions.
- UltiPlot® graphics display functions.
- UltiWord® word processing application.
- UltiWriter™ word processing application.

For more information on these applications, see the appropriate manuals, listed in the section Related Manuals.

For more information on the Ultimate Operating System, call your Ultimate representative, or call The Ultimate Corp. at (201) 887-9222.
How to Use This Manual

How the Manual is Organized

The manual consists of seven chapters and three appendices.

Chapter 1 is an introduction to Ultimate RECALL and explains the report format and software components of Ultimate RECALL.

Chapter 2 describes how to form an Ultimate RECALL sentence.

Chapter 3 describes the Ultimate RECALL system commands.

Chapter 4 describes attribute definition items.

Chapter 5 describes processing codes that can be used to convert your data to output format.

Chapter 6 describes the forms reporting features and how to use Ultimate RECALL to format and place your data.

Chapter 7 contains hints on using Ultimate RECALL.

Appendix A is a glossary of terms used in Ultimate RECALL.

Appendix B lists the attribute definition items that are supplied by default to every new account.

Appendix C shows all the elements of the database used throughout the manual in examples.
# Conventions

The following conventions are used in this manual:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UPPER CASE</strong></td>
<td>Characters in syntax definitions printed in upper case are required and must appear exactly as shown.</td>
</tr>
<tr>
<td><strong>lower case</strong></td>
<td>Characters in syntax definitions printed in lower case are parameters to be supplied by the user.</td>
</tr>
<tr>
<td><strong>{}</strong></td>
<td>Braces surrounding a parameter indicate that the parameter is optional and may be included or omitted at the user's option.</td>
</tr>
<tr>
<td><strong>RETURN</strong></td>
<td>The word RETURN indicates a physical carriage return pressed at the keyboard. A RETURN is required to complete a command line, and signals the system to begin processing the command.</td>
</tr>
<tr>
<td><strong>enter</strong></td>
<td>The word enter is used to mean &quot;type in the required text, then press RETURN.&quot;</td>
</tr>
<tr>
<td><strong>X'nn'</strong></td>
<td>This form is used to define a hexadecimal number where 'nn' is the hex value; for example, X'0B', X'41', X'FF'.</td>
</tr>
<tr>
<td><strong>Enter option</strong></td>
<td>This typeface is used for messages and prompts displayed by the system.</td>
</tr>
<tr>
<td><strong>CAUDIT</strong></td>
<td>This typeface in bold is used in examples to indicate user input.</td>
</tr>
<tr>
<td><strong>filename</strong></td>
<td>Any valid filename format can be used when filename is specified. Filename formats are described in the <em>Ultimate System Commands Guide</em>.</td>
</tr>
</tbody>
</table>
Related Manuals

The following is a list of the manuals that provide more information on topics described in this document. The document number next to each manual is to be used when ordering manuals.

For a complete list of Ultimate system manuals, or to order manuals, refer to Ultimate's Documentation Update brochure, or call Ultimate's administration department at (201) 887-9222.

<table>
<thead>
<tr>
<th>Manual</th>
<th>Document Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ultimate BASIC Language Reference Guide</td>
<td>6929-3</td>
</tr>
<tr>
<td>Beginner's Guide to Ultimate®</td>
<td>6977</td>
</tr>
<tr>
<td>Guide to the Ultimate Editors</td>
<td>6939</td>
</tr>
<tr>
<td>PROC Manual</td>
<td>6936 (Bound)</td>
</tr>
<tr>
<td></td>
<td>6967 (Looseleaf)</td>
</tr>
<tr>
<td>UltiKit User's Guide</td>
<td>6991</td>
</tr>
<tr>
<td>Ultimate System Commands Guide</td>
<td>6985</td>
</tr>
<tr>
<td>Ultimate System Management Guide</td>
<td>6960 (Bound)</td>
</tr>
<tr>
<td></td>
<td>6964 (Looseleaf)</td>
</tr>
<tr>
<td>Ultimate UPDATE User's Guide</td>
<td>6963</td>
</tr>
<tr>
<td>UltiPlot Reference Guide</td>
<td>6976 (Bound)</td>
</tr>
<tr>
<td></td>
<td>6980 (Looseleaf)</td>
</tr>
<tr>
<td>UltiPlot Training Guide</td>
<td>6975 (Bound)</td>
</tr>
<tr>
<td></td>
<td>6979 (Looseleaf)</td>
</tr>
<tr>
<td>UltiWord Reference Guide</td>
<td>6904 (Bound)</td>
</tr>
<tr>
<td></td>
<td>6905 (Looseleaf)</td>
</tr>
<tr>
<td>UltiWord Training Guide</td>
<td>6908 (Bound)</td>
</tr>
<tr>
<td></td>
<td>6948 (Looseleaf)</td>
</tr>
<tr>
<td>UltiWriter System Administrator's Guide</td>
<td>15121</td>
</tr>
<tr>
<td>UltiWriter Training Guide</td>
<td>15122</td>
</tr>
<tr>
<td>UltiWriter User's Guide</td>
<td>15120</td>
</tr>
<tr>
<td>Ultimate RECALL User's Guide</td>
<td>6972-1</td>
</tr>
</tbody>
</table>

Confidential and Proprietary to The Ultimate Corp.
1 Introduction

Ultimate RECALL is a general-purpose data retrieval language that enables you to selectively retrieve information from your database and create customized reports. It uses simple, English-like sentences to query the database, so you do not need any programming experience to develop a variety of reports. Ultimate RECALL statements can contain any number of variable-length words and allow flexibility in word order and syntax. This makes Ultimate RECALL an easy-to-use language even for novice Ultimate users.

You can perform the following functions with Ultimate RECALL:

- create reports at any time using TCL commands.
- select items to be processed by other system commands.
- sort by any number of attributes.
- use multivalued data.
- specify complex selection criteria.
- specify multiple levels of breaks and totals.
- use preprinted forms.
- print labels.
- call BASIC programs for special processing.
- reformat items in files.
- write selected information to tape.
- provide statistical information about files, items, and attributes.
- produce checksums for any attributes in your database.
**Introduction**

**Report Format**

The general format of an Ultimate RECALL report includes a heading line on each page, the report body, and a message line at the end of the report. See Figures 1-1 and 1-2.

The page width and length of the report are determined by the current setting of the terminal's TERM command. The most common settings for a terminal are 79 columns and 24 lines, and for a printer, 132 columns and 60 lines.

When the report is displayed on the terminal, Ultimate RECALL pauses at the bottom of each page, unless the NOPAGE modifier or N option has been specified. To continue to the next page, press either RETURN or the space bar.

**Heading Line**

The default heading line displays the date, time, and page number. The heading can be changed by a HEADING modifier, or can be omitted by specifying a HDR-SUPP or SUPP modifier or (H) option. (Modifiers and options are discussed in Chapter 2.)

**Report Body**

The report body contains the detail lines as specified in the statement. If forms expressions are used, the report body is formatted as specified by the forms expressions. If forms expressions are not used, the report body can be in columnar or non-columnar format, depending on the number of characters to be printed. If the sum of the column widths plus one for each separator between columns does not exceed the page width, a columnar report is output. If the requested output exceeds the page width, a non-columnar report is output, with one attribute name and its values per line.

In a columnar report, the headings are displayed across the top of the page. Each item is reported as a detail line, with its data fields displayed in their respective columns. The column headings are repeated at the top of each page. Column headings for dependent attributes contain an asterisk.

In a non-columnar report, the attribute headings are listed down the side of the report with their respective values immediately to the right, one
item at a time. Each attribute is displayed on a separate line. Each item is treated individually and a column heading is displayed only if the item being listed has values for that attribute. Dependent attributes are indented two spaces under their controlling attribute.

The column heading is taken from attribute 3; if attribute 3 is null, the item-ID of the attribute definition is used as the column heading. The column width is taken from attribute 10 or from the length of the column heading, whichever is longer. If a column heading is multivalued, each value is printed on a new heading line (multi-line column headings). Column headings can be omitted by specifying a COL-HDR-SUPP modifier or (C) option.

Figure 1-1 is an example of a columnar report. Figure 1-2 is an example of a non-columnar report.

Message Line

The message line shows the number of items on the report. It is displayed at the end of the report unless the Ultimate RECALL statement contains a HEADING or HDR-SUPP modifier or (H) option.
Figure 1-1. Columnar Report Format

Figure 1-2. Non-Columnar Report Format
Components of Ultimate RECALL

To use Ultimate RECALL, you need the following:

• Ultimate RECALL connectives in the form of master dictionary items. (Connectives are standard elements of Ultimate RECALL, such as BREAK-ON, WITH, or BY-EXP, that are used to specify formatting, selection criteria, or sort criteria. For a list of connectives, see Table 1-1.)

• Ultimate RECALL system commands. For a list of system commands, see Table 1-2.

• files with dictionaries that contain attribute definition items for data that is to be included in report.

The system commands and connectives are included in every new account by default. The Ultimate RECALL connectives are described in Chapter 2, except for END-WINDOW and WINDOW, which are described in chapter 6. The Ultimate RECALL system commands are described in Chapter 3.

Files and attribute definition items are created by applications. Attribute definition items are described in Chapter 4. There are also attribute definition items included in every new account that can be used with any file. For a list of these attribute definition items, see Appendix B.
<table>
<thead>
<tr>
<th>Connective</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>relational operator.</td>
</tr>
<tr>
<td>&amp;</td>
<td>relational operator.</td>
</tr>
<tr>
<td>&lt;</td>
<td>relational operator.</td>
</tr>
<tr>
<td>&lt;=</td>
<td>relational operator.</td>
</tr>
<tr>
<td>=</td>
<td>relational operator.</td>
</tr>
<tr>
<td>=&lt;</td>
<td>relational operator.</td>
</tr>
<tr>
<td>=&gt;</td>
<td>relational operator.</td>
</tr>
<tr>
<td>&gt;</td>
<td>relational operator.</td>
</tr>
<tr>
<td>&gt;=</td>
<td>relational operator.</td>
</tr>
<tr>
<td>A</td>
<td>throwaway modifier.</td>
</tr>
<tr>
<td>AFTER</td>
<td>relational operator.</td>
</tr>
<tr>
<td>AN</td>
<td>throwaway modifier.</td>
</tr>
<tr>
<td>AND</td>
<td>relational operator.</td>
</tr>
<tr>
<td>ANY</td>
<td>throwaway modifier.</td>
</tr>
<tr>
<td>ARE</td>
<td>throwaway modifier.</td>
</tr>
<tr>
<td>BEFORE</td>
<td>relational operator.</td>
</tr>
<tr>
<td>BREAK-ON</td>
<td>modifier.</td>
</tr>
<tr>
<td>BY</td>
<td>sort criteria.</td>
</tr>
<tr>
<td>BY-DSND</td>
<td>sort criteria.</td>
</tr>
<tr>
<td>BY-EXP</td>
<td>sort criteria.</td>
</tr>
<tr>
<td>BY-EXP-DSND</td>
<td>sort criteria.</td>
</tr>
<tr>
<td>COL-HDR-SUPP</td>
<td>modifier.</td>
</tr>
<tr>
<td>Connective</td>
<td>Usage</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>DATA</td>
<td>throwaway modifier.</td>
</tr>
<tr>
<td>DBL-SPC</td>
<td>modifier.</td>
</tr>
<tr>
<td>DET-SUPP</td>
<td>modifier.</td>
</tr>
<tr>
<td>EACH</td>
<td>selection criteria.</td>
</tr>
<tr>
<td>END-WINDOW</td>
<td>forms.</td>
</tr>
<tr>
<td>EQ</td>
<td>relational operator.</td>
</tr>
<tr>
<td>EVERY</td>
<td>selection criteria.</td>
</tr>
<tr>
<td>FILE</td>
<td>throwaway modifier.</td>
</tr>
<tr>
<td>FOOTING</td>
<td>modifier.</td>
</tr>
<tr>
<td>FOR</td>
<td>throwaway modifier.</td>
</tr>
<tr>
<td>GE</td>
<td>relational operator.</td>
</tr>
<tr>
<td>GRAND-TOTAL</td>
<td>modifier.</td>
</tr>
<tr>
<td>GT</td>
<td>relational operator.</td>
</tr>
<tr>
<td>HDR-SUPP</td>
<td>modifier.</td>
</tr>
<tr>
<td>HEADER</td>
<td>modifier.</td>
</tr>
<tr>
<td>HEADING</td>
<td>modifier.</td>
</tr>
<tr>
<td>ID-SUPP</td>
<td>modifier.</td>
</tr>
<tr>
<td>IF</td>
<td>selection criteria.</td>
</tr>
<tr>
<td>IN</td>
<td>throwaway modifier.</td>
</tr>
<tr>
<td>ITEMS</td>
<td>throwaway modifier.</td>
</tr>
<tr>
<td>LE</td>
<td>relational operator.</td>
</tr>
<tr>
<td>LIKE</td>
<td>relational operator.</td>
</tr>
</tbody>
</table>
Table 1-1. Connectives (3 of 3)

<table>
<thead>
<tr>
<th>Connective</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>L PTR</td>
<td>modifier.</td>
</tr>
<tr>
<td>LT</td>
<td>relational operator.</td>
</tr>
<tr>
<td>NE</td>
<td>relational operator.</td>
</tr>
<tr>
<td>NO</td>
<td>modifier.</td>
</tr>
<tr>
<td>NOPAGE</td>
<td>modifier.</td>
</tr>
<tr>
<td>NOT</td>
<td>relational operator.</td>
</tr>
<tr>
<td>OF</td>
<td>throwaway modifier.</td>
</tr>
<tr>
<td>ONLY</td>
<td>modifier.</td>
</tr>
<tr>
<td>OR</td>
<td>relational operator.</td>
</tr>
<tr>
<td>PAGE</td>
<td>throwaway modifier.</td>
</tr>
<tr>
<td>SUPP</td>
<td>modifier.</td>
</tr>
<tr>
<td>TAPE</td>
<td>modifier.</td>
</tr>
<tr>
<td>THE</td>
<td>throwaway modifier.</td>
</tr>
<tr>
<td>TOTAL</td>
<td>modifier.</td>
</tr>
<tr>
<td>USING</td>
<td>modifier.</td>
</tr>
<tr>
<td>WINDOW</td>
<td>forms.</td>
</tr>
<tr>
<td>WITH</td>
<td>selection criteria.</td>
</tr>
<tr>
<td>WITHIN</td>
<td>modifier.</td>
</tr>
<tr>
<td>WITHOUT</td>
<td>selection criteria.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CHECK-SUM</td>
<td>generates a checksum of selected items.</td>
</tr>
<tr>
<td>COUNT</td>
<td>outputs the total number of selected items.</td>
</tr>
<tr>
<td>HASH-TEST</td>
<td>produces statistics showing the distribution of items based on a test modulo.</td>
</tr>
<tr>
<td>ISTAT</td>
<td>produces statistics showing the distribution of items based on the current modulo.</td>
</tr>
<tr>
<td>LIST</td>
<td>outputs selected items and attributes in a formatted report, based on the attribute definition items.</td>
</tr>
<tr>
<td>LIST-ITEM</td>
<td>lists contents of selected items.</td>
</tr>
<tr>
<td>LIST-LABEL</td>
<td>outputs selected items and attributes in a label format; prompts for parameters such as number of items to print across the report page.</td>
</tr>
<tr>
<td>REFORMAT</td>
<td>produces a new file from selected items and attributes, on disk or tape.</td>
</tr>
<tr>
<td>S-DUMP</td>
<td>outputs selected and sorted file items to tape.</td>
</tr>
<tr>
<td>SELECT</td>
<td>produces a select-list that contains selected item-ids and, optionally, specified attributes.</td>
</tr>
<tr>
<td>SORT</td>
<td>outputs selected items and attributes, sorted by specified attribute values, in a formatted report, based on the attribute definition items.</td>
</tr>
<tr>
<td>SORT-ITEM</td>
<td>lists contents of selected items, sorted by specified attribute values.</td>
</tr>
<tr>
<td>SORT-LABEL</td>
<td>outputs selected items and attributes, sorted by specified attributes, in a label format; prompts for parameters such as number of items to print across the report page.</td>
</tr>
</tbody>
</table>

Table 1-2. Ultimate RECALL Commands (1 of 2)
Table 1-2. Ultimate RECALL Commands (2 of 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SREFORMAT</td>
<td>produces a new file from selected items and attributes, sorted by specified attribute values, on disk or tape.</td>
</tr>
<tr>
<td>SSELECT</td>
<td>produces a select-list that contains selected item-ids and, optionally, specified attributes, sorted by specified attribute values.</td>
</tr>
<tr>
<td>STAT</td>
<td>outputs the total value of a specified attribute in all selected items; also outputs the total number of items selected and the average value of the attribute per item.</td>
</tr>
<tr>
<td>SUM</td>
<td>outputs the total value of a specified attribute in all selected items.</td>
</tr>
<tr>
<td>T-DUMP</td>
<td>outputs selected items to tape.</td>
</tr>
<tr>
<td>T-LOAD</td>
<td>loads selected or specified file items from the attached tape.</td>
</tr>
</tbody>
</table>
2 Forming an Ultimate RECALL Statement

An Ultimate RECALL statement requires only an Ultimate RECALL command and a filename. This produces a report that includes all data items in the file. Other parameters can be included in the statement to limit the report to selected items or to include specified attributes.

The following is the general syntax for an Ultimate RECALL statement; each element of the syntax is described in detail in this chapter. The commands that use this syntax are described in Chapter 3.

Syntax

command filename {itemlist} {sel-criteria} {sort-criteria} {output-specifications} {print-limiters} {modifiers} {options}

command

any Ultimate RECALL command defined in the current account’s Master Dictionary (MD). For information on these commands, see Chapter 3, Ultimate RECALL Commands.

filename

name of file to be used.

itemlist

specifies items to select for processing consideration; item-IDs should be enclosed in single quotes (''). If no itemlist is present, all items are considered, unless a select-list has been created by the previous statement. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.

sel-criteria

provides the tests that the items chosen for consideration must pass in order to be included in the Ultimate RECALL report. More than one selection criterion can be specified in order to select only those items that meet a certain set of criteria. If no selection criteria are present, all items being considered are selected.

sort-criteria

specifies an attribute name to use as the sort key to resequence the items for the report output; sort criteria is used only with sort commands. A number of sort criteria can be used in order to create a multi-level sorted
report. The sort can be in ascending or descending order. If no sort criteria are present in a sort statement, the items are sorted by item-ID in ascending order.

**output-specifications**

indicates which attribute definition items are to be used; these determine which fields of information are included on the report, and how they are formatted.

**print-limiters**

indicates criteria that a value in the specified attribute must meet in order to be included on the report. If no print limiters are present, all the values for a multivalued attribute are included. Print limiters are usually associated with multivalued attributes only.

**modifiers**

control the format and layout of the report, such as double-spacing, handling control breaks and totals, suppressing item-IDs, adding headings or footings, etc. If no modifiers or options are present, the report is single spaced without any breaks or totals, and item-IDs are automatically included in the first column of the report under a column heading of `filename`.

**options**

control the format and layout of the report, such as double-spacing, or suppressing item-IDs. If no options are present, the report is single spaced and item-IDs are automatically included in the first column of the report under a column heading of `filename`.

Relational operators, such as = or NOT, can be used in an itemlist, with selection criteria, and with print limiters to further refine the report.

**Description**

An Ultimate RECALL statement can be executed by entering the statement at the system (TCL) level. Complex Ultimate RECALL statements are often put into PROCs for ease of editing and running.

Spaces are used as delimiters (separators) between elements of the Ultimate RECALL statement. However, if elements are enclosed in single quotes, double quotes, or backslashes, they do not need to be separated by spaces.
All elements used in the statement, except for literal values, must exist in the file's dictionary or the account's MD. If an element is not found in the file dictionary, the account's MD is searched. If it is still not found, Ultimate RECALL creates a new element by concatenating the unknown element to the next element, separating them with a space. The file dictionary and the master dictionary are again searched using the new element. This concatenating continues until a match is found, or until the end of the statement is reached. If the end of the statement is reached, and the element is still not found, an error message similar to the following, starting with the unknown element, is displayed:

[24] The word "element1 element2 element3..." cannot be identified.

Elements that are used as relational operators, or to form selection criteria, sort criteria, and output specifications are often referred to as connectives. Connectives are defined in the account's MD. Table 1-1 in Chapter 1 lists the Ultimate RECALL connectives, which are described in detail in this chapter under their usage.

**Note:** Output specifications indicate the attributes to be output. The output specifications can also be used to give the exact position on the page of each attribute's data value, as well as to include literal data such as header information. In this case, the output specifications are called forms expressions and the output format is called forms output. This capability can be useful, for example, when printing on preprinted forms.

For information on forms, see Chapter 6, Using Ultimate RECALL with Report Forms.
Relational Operators

Relational operators, such as = or NOT, are used to specify a range of values or to exclude specific items. Relational operators can be used in itemlists, selection criteria, and print and BY-EXP limiters. Table 2-1 lists the relational operators.

If relational operators are used in the itemlist, the itemlist is said to be complex. If specific values are entered, they must be enclosed in double quotes or backslashes. The value applies to the previous attribute name.

If a statement contains both AND and OR operators, expressions connected with AND are evaluated before those with OR; that is, the precedence of evaluation is AND over OR. Operators are evaluated from left to right. For example, the following expression selects items with NAME that start with characters whose ASCII value is less than K, or that start with characters whose ASCII value is greater than M but less than S:

\[
\text{WITH NAME} < "K" \text{ OR } "M" \text{ AND } "S"
\]

If two pairs are specified with neither AND nor OR specified, OR is used. For example, the first statement selects items with dates either before 07/01/92 or after 08/15/92, whereas the second statement selects items with dates between 07/01/92 and 08/15/92:

\[
\begin{align*}
\text{WITH DATE} & \ < "07/01/92" \text{ OR } "08/15/92" \text{ DATE COMPANY} \\
\text{WITH DATE} & \ \ge "07/01/92" \text{ AND LE } "08/15/92" \text{ DATE COMPANY}
\end{align*}
\]

If no operator is specified, = is assumed. For example, the following selects items with item-IDS less than 1685 or equal to 1696.

\[
\text{INVOICE} < "1685" \text{ OR } "1696"
\]

If you specify a relational operator with an item-ID, all item-IDS are accessed and compared, whereas if you just specify the item-ID, only that item is accessed. The latter method is much faster when working with large files, and should be used whenever possible. For example, the first statement looks at all items in the INVOICE file, whereas the second statement looks at only the specified item. The same item is retrieved in both cases.

\[
\begin{align*}
\text{LIST INVOICE} & \ = "1682" \\
\text{LIST INVOICE} & \ '1682'
\end{align*}
\]
### Table 2-1. Relational Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQ</td>
<td>equal to.</td>
</tr>
<tr>
<td>=</td>
<td></td>
</tr>
<tr>
<td>GT</td>
<td>greater than.</td>
</tr>
<tr>
<td>&gt;</td>
<td></td>
</tr>
<tr>
<td>AFTER</td>
<td></td>
</tr>
<tr>
<td>LT</td>
<td>less than.</td>
</tr>
<tr>
<td>&lt;</td>
<td></td>
</tr>
<tr>
<td>BEFORE</td>
<td></td>
</tr>
<tr>
<td>GE</td>
<td>greater than or equal to.</td>
</tr>
<tr>
<td>&gt;=</td>
<td></td>
</tr>
<tr>
<td>=&gt;</td>
<td></td>
</tr>
<tr>
<td>LE</td>
<td>less than or equal to.</td>
</tr>
<tr>
<td>&lt;=</td>
<td></td>
</tr>
<tr>
<td>=&lt;</td>
<td></td>
</tr>
<tr>
<td>NE</td>
<td>not equal to.</td>
</tr>
<tr>
<td>#</td>
<td></td>
</tr>
<tr>
<td>NOT</td>
<td></td>
</tr>
<tr>
<td>LIKE</td>
<td>search for items that sound like; for more</td>
</tr>
<tr>
<td></td>
<td>information, see next subsection.</td>
</tr>
<tr>
<td>AND</td>
<td>both conditions must be true.</td>
</tr>
<tr>
<td>&amp;</td>
<td></td>
</tr>
<tr>
<td>OR</td>
<td>one or both conditions must be true.</td>
</tr>
</tbody>
</table>
Using Ultimate RECALL

<table>
<thead>
<tr>
<th>Invoice Number</th>
<th>Date</th>
<th>Company Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1681</td>
<td>08/23/92</td>
<td>Kelley Brothers</td>
</tr>
<tr>
<td>1682</td>
<td>08/24/92</td>
<td>Kelley Brothers</td>
</tr>
<tr>
<td>1686</td>
<td>08/21/92</td>
<td>Quality Lighting Products</td>
</tr>
<tr>
<td>1687</td>
<td>06/16/92</td>
<td>Service Office Products</td>
</tr>
</tbody>
</table>

4 items listed.

<table>
<thead>
<tr>
<th>Invoice Number</th>
<th>Date</th>
<th>Company Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1683</td>
<td>08/09/92</td>
<td>Service Office Products</td>
</tr>
<tr>
<td>1696</td>
<td>07/01/92</td>
<td>Universal Copiers</td>
</tr>
</tbody>
</table>

2 items listed.

<table>
<thead>
<tr>
<th>Invoice Number</th>
<th>Company Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1681</td>
<td>Kelley Brothers</td>
</tr>
<tr>
<td>1682</td>
<td>Kelley Brothers</td>
</tr>
<tr>
<td>1683</td>
<td>Service Office Products</td>
</tr>
<tr>
<td>1696</td>
<td>Universal Copiers</td>
</tr>
</tbody>
</table>

4 items listed.

<table>
<thead>
<tr>
<th>Invoice Number</th>
<th>Contact Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1681</td>
<td>Jerry</td>
</tr>
<tr>
<td>1682</td>
<td>Jerry</td>
</tr>
<tr>
<td>1686</td>
<td>Mat</td>
</tr>
<tr>
<td>1696</td>
<td>Marina</td>
</tr>
</tbody>
</table>

4 items listed.
LIKE Operator

The LIKE relational operator provides a soundex capability. This capability can be useful for cross references based on words or names that sound alike. Soundex codes also overcome problems with upper case and lower case, typographical errors, and misspellings in a database.

When the LIKE operator is specified, Ultimate RECALL converts the specified string to a soundex code. The soundex code is made up a maximum of four values. The first value in the soundex code is the first alphabetic character in the string. Subsequent values in the soundex codes are numeric values given to consonants. Case is ignored. Non-alphabetic characters are ignored. Wild cards are treated as any other non-alphabetic character; that is, they are ignored. If two or more characters with the same numeric value are adjacent, only one value is returned.

The LIKE operator selects all attributes with soundex codes that match the specified string. Words with a similar arrangement of consonants have similar soundex codes, regardless of the actual spelling. Also, similar sounding consonants may have the same soundex code.

The following sets of letters match each other:

- a e i o u h w y
- b f p v
- c g j k q s x z
- d t
- l
- m n
- r

The only letter in the data that must match the soundex code exactly is the first letter. For example, the following words all have the same soundex code and therefore all match the parameter LIKE "lorry":

Laura  Lora  Laurie  lorry
Lorrie  Lori  LARRY
Using Ultimate RECALL

:SORT INVOICE WITH NAME LIKE "JERI" COMPANY NAME

<table>
<thead>
<tr>
<th>PAGE</th>
<th>1</th>
</tr>
</thead>
</table>

INVOICE... Company Name........ Contact..............

<table>
<thead>
<tr>
<th></th>
<th>Kelley Brothers</th>
<th>Jerry</th>
</tr>
</thead>
<tbody>
<tr>
<td>1681</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1682</td>
<td>Kelley Brothers</td>
<td>Jerry</td>
</tr>
</tbody>
</table>

2 items listed.
Wild Cards are used to specify a partial search string instead of a complete search string. A wild card matches a single character or any number of characters, depending on the wild card used.

Wild cards can be used with a complex itemlist (that is, one that contains a relational operator), with selection criteria, and with print limiters. However, a wild card will not work in an Ultimate RECALL statement if the attribute has an MR conversion code (attribute 7 of the attribute definition item) and the attribute has pad characters on the left. (The MR code is described in Chapter 5.)

Table 2-2 lists the wild card characters.

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>...^...</td>
<td>caret: matches any single character.</td>
</tr>
<tr>
<td>...]</td>
<td>right bracket: matches all characters to the right of the specified string.</td>
</tr>
<tr>
<td>[...]</td>
<td>left bracket: matches all characters to the left of the specified string.</td>
</tr>
<tr>
<td>[...]</td>
<td>left and right brackets; match all attributes or item-IDs that contain the specified string.</td>
</tr>
</tbody>
</table>
Using Ultimate RECALL

SORT INVOICE = "RD]"
right bracket; match all characters to the right of the specified string; that is, find all items with item-IDs that begin with the characters RD.

SORT INVOICE = "[82"
left bracket; match all characters to the left of the specified string; that is, find all items with item-IDs that end with the characters 82.

SORT INVOICE WITH NAME = "[Jerry]"
left and right brackets; match all attributes that contain the specified string; that is, find all items where the attribute NAME contains the characters Jerry.

SORT INVOICE WITH PROD.NO = "80^6"
caret; match any single character; that is, find all items with the attribute PROD.NO of the form 80^6, where ^ is any character.
When a file is listed, the default listing contains the item-IDs, with the filename as the default heading. Each item-ID is displayed on a single line; item-IDs do not fold if they exceed the column width. If other attributes are listed, they will overlay the item-IDs that exceed the column width.

Attributes 7, 9, and 10 in a file definition item are used to sort, select, and format the item-ID and follow the same rules as the corresponding attributes in attribute definition items. The conversion code in attribute 7 is used for output. The justification code in attribute 9 is used for sorting and output. The length value in attribute 10 is used for output. Because item-IDs do not fold if they exceed the column width, the length is important if item-IDs are often longer than the filename (used as the column heading).

If an item in the account's MD is a synonym file definition item (Q-pointer), attributes 7, 9, and 10 of the file definition item pointed to by the Q-pointer are used.

Correlative specifications (attribute 8) in file definition items and Q-pointers are ignored unless a WITHIN connective is used and the correlative is a V processing code (WITHIN is described in the section Modifiers, later in this chapter).

For more information on defining attributes 7, 9, and 10, see Chapter 4, Attribute Definition Items.
Itemlists

Itemlists specify the item-IDs that are to be considered by the command. If no itemlist is present, all items in the file are considered. An itemlist can be explicitly named in the statement or it can be implicit, that is, taken from a select list. After the items are selected, the final item selections for inclusion on the report are determined by the selection-criteria.

Ultimate RECALL always uses an explicit itemlist, if present, to process a statement.

Wild cards and relational operators can be used with the item-IDs in an explicit itemlist to specify a range of items or to exclude specific items. If relational operators are used in the itemlist, the itemlist is said to be complex. When resolving a complex itemlist, Ultimate RECALL compares every item-ID in the file to each itemlist component in the statement in order to select a list of items. The type of comparison (numeric or alphabetic) depends on how the item-ID is sorted. For information on sorting, see the discussion of V/TYP in Chapter 4.

Each item-ID in the itemlist can be preceded by a relational operator.

If a relational operator is specified with an item-ID, all item-IDs in the file are accessed, whereas if just the item-ID is specified, only that item is accessed. The latter method is much faster when working with large files, and should be used whenever possible.

Explicit Itemlist

If the item-IDs of the items to be considered are explicitly named in the Ultimate RECALL statement, the itemlist is said to be explicit. In general, if explicit item-IDs are specified, they should be enclosed in single quotes ('). However, if explicit item-IDs are specified immediately following the filename, they can be enclosed in double quotes (") or backslashes (\\). Any number of items can be specified.

Implicit Itemlist

If no item-IDs are included in the Ultimate RECALL statement and if a select-list is present, the itemlist is said to be implicit. (Select lists are described in the next subsection.)
Select-Lists

A select-list is present if a SELECT, SSELECT, or QSELECT command was the last command processed, or if a saved select-list was just retrieved by a GET-LIST command. The number of items in the resulting list is indicated by the message:

n items selected.

When a select-list is available and no explicit itemlist is specified, the items in the select-list are used by the Ultimate RECALL command. The items are subject to the selection criteria in order to determine final item selection.

A select-list can be saved by storing the list immediately after it has been created. Lists are always stored in the DICTIONARY-FILE. (Before creating a select-list that you want to save, be sure you have defined a DICTIONARY-FILE for your account.) You can list the stored select-lists for your account by using the following statement:

SORT ONLY DICTIONARY-FILE

To save a list, use the following command immediately after the list was created:

SAVE-LIST list-name

To edit a saved select-list, use the following command:

EDIT-LIST list-name

To delete a saved select-list, use the following command:

DELETE-LIST list-name

For more information on these commands, see the Ultimate System Commands Guide.

The elements in a select-list can be used to match item-IDs in any data file, not just the file from which the select-list was created. For example, a select-list from AFFILE can be the itemlist used by a command that references BFILE. The command accesses and lists the items in BFILE with the same item-IDs as the items selected from AFFILE.

The following notes may be helpful in using select-lists with other Ultimate software:
Using Ultimate RECALL

BASIC  The READNEXT statement makes an existing select-list available to the BASIC program. An external select list can be generated by the EXECUTE statement from within BASIC, or the select list can be generated just before executing the BASIC program. An internal select-list can be generated by using the SELECT statement. An external select-list overrides the internal select list created by the first SELECT statement in a BASIC program.

UltiWord  The /SELECT, /SSELECT, and /QSELECT commands can be used to generate select lists. The /READNEXT statement makes the select-list available to UltiWord as described in the UltiWord Reference Guide.

UltiWriter  The @FN and @GL codes retrieve previously saved select-lists and use the item-IDs as described in the UltiWriter User's Guide.

System Commands

The select-list is available to any system command as an implicit itemlist. If the command contains an explicit itemlist, that list overrides the select-list. If the command is an Ultimate RECALL-type command with selection criteria, the selection criteria are applied to items named in the select-list.
Selection Criteria

Selection criteria are conditions that attribute values in an item must meet in order for that item to be included on the Ultimate RECALL report. The selection criteria tests are applied to all file items if no itemlist exists, or to all those that have met the requirements of an itemlist. After the selection criteria has been applied, all items that meet the criteria are included on the report. This clause specifies the acceptable values an attribute must have in order to select that item.

An Ultimate RECALL statement can contain any number of selection criteria. Criteria can be separated by either the AND or OR operator. If neither is specified, OR is assumed.

Syntax

WITH|IF {NO|NOT} {EVERY|EACH} attrib-name {rel-op} {value-list}
WITHOUT {EVERY|EACH} attrib-name {rel-op} {value-list}

WITH specifies selection criteria; WITH and IF are synonymous and can be used interchangeably.

IF excludes items that have the specified values; NO and NOT are synonymous and can be used interchangeably.

NO

NOT

WITHOUT excludes items that have the specified values. This is equivalent to WITH NO.

EVERY used for multivalued attributes to specify that all subvalues must have the specified values; EVERY and EACH are synonymous and can be used interchangeably.

EACH

attrib-name name of the attribute whose value is being tested.

rel-op relational operator. If omitted, the equality (EQ) operator is assumed.

value-list specifies the legal values for the attribute to match in order to meet the selection criterion. The values must be enclosed in double quotes (".

Description

The simplest selection criterion is of the following form, which means that the item is selected if the specified attribute (attrib-name) has at least one value (is not null):

WITH|IF {NO|NOT} {EVERY|EACH} attrib-name {rel-op} {value-list}
Complex selection criteria are made up of more than one criterion; for example, the following example has complex selection criteria. An item is selected if attributes A1 and A2 are not null, or if the value of attribute A3 is "GO".

WITH A1 AND WITH A2 OR WITH A3 = "GO"

Relational operators and wild cards can be used with selection criteria.

:LIST INVOICE COMPANY NAME WITH COMPANY = "KELLEY"

INVOICE... Company Name........ Contact
1681    Kelley Brothers    Jerry
1682    Kelley Brothers    Jerry
2 items listed.

:SORT INVOICE WITH DATE < "08/01/92" AND WITH FLAGALL = "0" COMPANY DATE TOT.PRICE
sorts all items where the value of the first delivery date is less than 08/01/92.

INVOICE... Company Name........... Invoice.. Tot Price.
Date
1696    Universal Copiers 07/01/92   $55.80

End of list
Selection Criteria

<table>
<thead>
<tr>
<th>LIST INVOICE DEL.DATE NAME COMPANY WITH EVERY DEL.DATE BEFORE &quot;08/01/92&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>lists invoices with all delivery dates before 08/01/92. (DEL.DATE is</td>
</tr>
<tr>
<td>dependent on PROD.NO and is not listed, but the items are selected.)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PAGE</th>
<th>INVOICE... Contact.......... Company Name..........</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13:21:30 09 JUL 1992</td>
</tr>
</tbody>
</table>

1683   Shelby       Service Office Products
1687   Lee          Service Office Products
1696   Marina       Universal Copiers

3 items listed.
## Sort Criteria

Sort criteria are specifications for sorting items so that an Ultimate RECALL report can be arranged in the desired sequence. After sorting, Ultimate RECALL uses the output specifications to produce the report.

Sort criteria are specified by preceding an attribute name with one of the four sort connectives.

### Syntax

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BY</strong> attrib-name</td>
<td>sorts items according to the ascending value (lowest to highest) of the specified attribute.</td>
</tr>
<tr>
<td><strong>BY-DSND</strong> attrib-name</td>
<td>sorts items according to the descending value (highest to lowest) of the specified attribute.</td>
</tr>
<tr>
<td><strong>BY-EXP</strong> attrib-name {exp-limiter}</td>
<td>explodes each multivalue into its own item for the report and sorts in ascending order.</td>
</tr>
<tr>
<td><strong>BY-EXP-DSND</strong> attrib-name {exp-limiter}</td>
<td>explodes each multivalue into its own item for the report and sorts in descending order.</td>
</tr>
</tbody>
</table>

**attrib-name** name of attribute by which to sort.

**exp-limiter** limits the explosion to values that meet the limit conditions; **exp-limiter** has the following form:

```plaintext
{ {rel-op} value-list }
```

- **rel-op** relational operator; if omitted, the equality (EQ) operator is assumed.
- **value-list** specifies the values each multivalue must match in order to be exploded. The values must be enclosed in double quotes (""), or backslashes (\).

### Description

Sorting is performed on intermediate format values. This means that correlatives are applied before sorting, but conversions are not. The order of the sort depends on the V/TYP in the attribute definition item. For more information on sort order, see Chapter 4.
Sort criteria are applied to the items that have been selected according to the values of the specified attributes. An Ultimate RECALL sort statement can contain any number of sort criteria.

The following sort commands accept sort criteria:

- S-DUMP
- SORT
- SORT-ITEM
- SORT-LABEL
- SPIE
- SPLOT
- SREFORMAT
- SSELECT

The sort commands are described in detail in Chapter 3.

If no sort criteria are present in the sort statement, Ultimate RECALL uses the item-ID as a sort key and sorts in ascending item-ID order.

To produce a report in descending item-ID sequence, you must set up an attribute definition item for the item-ID, which is attribute 0. For example, the following defines the item-ID for the INVOICE file:

```
item-ID INVOICE#
  001 A
  002 0
  003 Invoice Number
  ...
  009 L
  010 10
```

To produce a report in descending order by invoice number, you could use a statement similar to the following:

```
:SORT INVOICE BY-DSND INVOICE# COMPANY
```
Using Ultimate RECALL

Multivalued Attributes

If the attribute being sorted is multivalued, only the first value is considered in the sort if the BY or BY-DSND connective is used. To sort on each of the multivalues, use the BY-EXP or BY-EXP-DSND connective, which causes each value in the specified attribute of the selected items to be exploded and sorted as a separate item.

If multiple BY-EXPs are specified in the statement, the attribute with the most multiple values is used to create the items. The other fields are treated as null if there is no data for all the values.

To limit the explosion to certain values, use the optional exp-limiters. The exp-limiters cause each value in the specified attribute to be tested against conditions. Only the values that meet the conditions are included in the sort and output.
:SORT INVOICE BY-EXP PROD.NO PROD.NO DESC '1681' '1682'

explodes the product number and creates one line for every product number.

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1681</td>
<td>0005 Herb Tea</td>
</tr>
<tr>
<td>1681</td>
<td>2025 Regular Tea</td>
</tr>
<tr>
<td>1682</td>
<td>5011 Hot Chocolate Regular</td>
</tr>
<tr>
<td>1682</td>
<td>5015 Hot Chocolate Sugar Free</td>
</tr>
<tr>
<td>1681</td>
<td>6032 Orange Juice</td>
</tr>
<tr>
<td>1682</td>
<td>6032 Orange Juice</td>
</tr>
<tr>
<td>1682</td>
<td>6065 Cranberry Juice</td>
</tr>
<tr>
<td>1681</td>
<td>6068 Grapefruit Juice</td>
</tr>
<tr>
<td>1682</td>
<td>6068 Grapefruit Juice</td>
</tr>
<tr>
<td>1681</td>
<td>7001 Water</td>
</tr>
<tr>
<td>1681</td>
<td>7015 Raspberry Soda</td>
</tr>
<tr>
<td>1681</td>
<td>7017 Wild Cherry Soda</td>
</tr>
<tr>
<td>1681</td>
<td>7055 Diet Cola</td>
</tr>
<tr>
<td>1682</td>
<td>7055 Diet Cola</td>
</tr>
<tr>
<td>1681</td>
<td>7056 Regular Cola</td>
</tr>
<tr>
<td>1681</td>
<td>7065 Diet Root Beer</td>
</tr>
<tr>
<td>1681</td>
<td>7066 Root Beer</td>
</tr>
<tr>
<td>1681</td>
<td>7331 Tomato Juice</td>
</tr>
<tr>
<td>1681</td>
<td>7345 Apple Juice</td>
</tr>
<tr>
<td>1681</td>
<td>8036 Cookies</td>
</tr>
<tr>
<td>1681</td>
<td>8123 Crackers</td>
</tr>
</tbody>
</table>

21 items listed.

:SORT INVOICE BY-EXP PROD.NO > "8000" PROD.NO DESC '1681' '1682'

limits the explosion to product numbers greater than 8000.

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1681</td>
<td>8036 Cookies</td>
</tr>
<tr>
<td>1681</td>
<td>8123 Crackers</td>
</tr>
</tbody>
</table>

2 items listed.
If a BY-EXP modifier is used in a SSELECT statement, each attribute in the select list will have both the item-ID and the value-number of each multivalue in an exploded item. The value number is in hexadecimal.

Lists produced by a SSELECT statement with a BY-EXP modifier can be used with the LIST command or in applications, where the value number can be used to access the required data.

The following example shows a select list produced by including a BY-EXP modifier in a SSELECT statement.

```
:SSELECT INVOICE BY-EXP PROD.NO '1681' '1682'
21 items selected.
:SSAVE-LIST SSEL
'SSEL' saved - 1 frames used.
:EDIT-LIST SSEL
select list contains the item-IDs and value number of the location of the product number.

EOI 02.
```
Output Specifications

Output specifications consist of attribute names that enumerate individual attributes to be included on an Ultimate RECALL report. The attributes are displayed in the order they are specified, except for controlling and dependent attributes, which have special requirements. (For more information on controlling and dependent attributes; see the next subsection.)

There are two connectives that can be used with output specifications: BREAK-ON and TOTAL. The general formats of these output specifications are:

```
BREAK-ON attrib-name
TOTAL attrib-name
```

BREAK-ON and TOTAL are described in the section, Modifiers, later in this chapter.

If the Ultimate RECALL statement contains no output specifications, Ultimate RECALL looks in the dictionary of the file for default output specifications, which are attributes defined by attribute definition items with consecutively numbered item-IDs starting with 1.

If no output specifications are present and there are no default output specifications, the minimum default report is produced. The minimum report contains only the item-IDs of the selected file items. (This same report format can be produced with the ONLY modifier.)

If there are default output specifications, the report will contain a column for each attribute referenced by these numbered item-IDs with the columns in numeric order. Numbers cannot be skipped. For example, if there is no 4, then even if there is a 5, the report stops at 3 and 5 is not listed.

To suppress printing of one or more attributes, but continue with the search for consecutively numbered item-IDs, the attribute definition items for attributes to skip can be given a special code of X in attribute 1 (instead of A). This does not break the sequence; the attribute is not listed on a default Ultimate RECALL report, but Ultimate RECALL searches for the next sequential item-ID.
Controlling and Dependent Attributes

In order to display a dependent attribute, its controlling attribute must also be specified.

Dependent attributes are always listed immediately following their controlling attribute. For example, the following example has three attribute names as output specifications. QTY is dependent on PROD.NO and so is displayed following it although it is specified first in the statement.
### Output Specifications

```plaintext
:LIST INVOICE '1682' COMPANY QTY PROD.NO

PAGE 1 08:56:34 09 JUL 1992

INVOICE... Company Name....... Product... Qty.
Number

<table>
<thead>
<tr>
<th>1682</th>
<th>Kelley Brothers</th>
<th>7055</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>6032</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6065</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6068</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5011</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5015</td>
<td>3</td>
</tr>
</tbody>
</table>

End of list
```
**Print Limiters**

Print limiters restrict the printing of specified attributes to only those values that meet the limit conditions. A print limiter consists of a relational operator and a value list, and is an optional part of output specifications.

**Syntax**

```
attrib-name {{rel-op} value-list}
```

- **attrib-name**: name of the attribute whose value is being tested.
- **rel-op**: relational operator. If omitted, the equality (EQ) operator is assumed.
- **value-list**: specifies the legal values for the attribute to match in order for a value to be printed. The values must be enclosed in double quotes (") or backslashes (\).

**Description**

If a print limiter is specified for a controlling attribute, dependent values associated with the controlling attribute values that fail to meet the limit condition are also suppressed.

Wild cards can be used with print limiters.
:SORT INVOICE PHONE = "714" COMPANY

displays only phone numbers that begin with 714, but displays all company names.

<table>
<thead>
<tr>
<th>PAGE 1</th>
<th>13:52:58 18 MAR 1992</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVOICE... Phone Number Company Name........</td>
<td></td>
</tr>
<tr>
<td>1681 714/555 6789 Kelley Brothers</td>
<td></td>
</tr>
<tr>
<td>1682 714/555 6789 Kelley Brothers</td>
<td></td>
</tr>
<tr>
<td>1683 Service Office Products</td>
<td></td>
</tr>
<tr>
<td>1684 Quality Lighting Products</td>
<td></td>
</tr>
<tr>
<td>1685 714/555 1234 Service Office Products</td>
<td></td>
</tr>
<tr>
<td>1696 Universal Copiers</td>
<td></td>
</tr>
</tbody>
</table>

6 items listed.

:SORT INVOICE PROD.NO = "60" DESC QTY PRICE

displays only PROD.NOS that begin with 60. Because DESC, QTY, and PRICE depend on PROD.NO, they are displayed only if PROD.NO is displayed.

<table>
<thead>
<tr>
<th>PAGE 1</th>
<th>14:00:11 18 MAR 1992</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVOICE... Product... Description.... Price..... Qty.</td>
<td></td>
</tr>
<tr>
<td>Number</td>
<td>*</td>
</tr>
<tr>
<td>1681</td>
<td>6032</td>
</tr>
<tr>
<td></td>
<td>4068</td>
</tr>
<tr>
<td>1682</td>
<td>6032</td>
</tr>
<tr>
<td></td>
<td>4068</td>
</tr>
<tr>
<td>1683</td>
<td>4032</td>
</tr>
<tr>
<td></td>
<td>4068</td>
</tr>
<tr>
<td>1686</td>
<td>5024</td>
</tr>
<tr>
<td></td>
<td>6032</td>
</tr>
<tr>
<td>1687</td>
<td></td>
</tr>
<tr>
<td>1696</td>
<td></td>
</tr>
</tbody>
</table>

6 items listed.
Modifiers

Modifiers can be used to specify the report source or destination, change the format, or to further specify selection criteria. The functions of some of these modifiers can also be specified as options in an Ultimate RECALL statement.

The standard Ultimate RECALL modifiers and options are part of the user account vocabulary that is supplied in the account's Master Dictionary (MD) when the account is created.

Table 2-3 lists the modifiers.
Table 2-3. Modifiers

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BREAK-ON</td>
<td>breaks a report into sets of items based on the value of an attribute.</td>
</tr>
<tr>
<td>COL-HDR-SUPP</td>
<td>suppresses the column heading line, and also page heading and end message.</td>
</tr>
<tr>
<td>DBL-SPC</td>
<td>inserts a blank line between detail lines.</td>
</tr>
<tr>
<td>DET-SUPP</td>
<td>suppresses the display of all detail lines.</td>
</tr>
<tr>
<td>FOOTING</td>
<td>defines a footing.</td>
</tr>
<tr>
<td>GRAND-TOTAL</td>
<td>formats the grand total line.</td>
</tr>
<tr>
<td>HDR-SUPP</td>
<td>suppresses the page heading and end message.</td>
</tr>
<tr>
<td>HEADER</td>
<td>defines a heading.</td>
</tr>
<tr>
<td>HEADING</td>
<td>defines a heading.</td>
</tr>
<tr>
<td>ID-SUPP</td>
<td>suppresses the display of item-IDs.</td>
</tr>
<tr>
<td>LPTR</td>
<td>routes the report output to the printer.</td>
</tr>
<tr>
<td>NOPAGE</td>
<td>omits the wait at the end of each page.</td>
</tr>
<tr>
<td>ONLY</td>
<td>suppresses the use of the default attribute definition items.</td>
</tr>
<tr>
<td>SUPP</td>
<td>suppresses the page heading and end message.</td>
</tr>
<tr>
<td>TAPE</td>
<td>causes the data to be retrieved from tape.</td>
</tr>
<tr>
<td>TOTAL</td>
<td>accumulates a total value.</td>
</tr>
<tr>
<td>USING</td>
<td>selects and formats items in one file using attribute definitions from another file.</td>
</tr>
<tr>
<td>WITHIN</td>
<td>selects items from list contained in specified item.</td>
</tr>
<tr>
<td>Throwaway</td>
<td>make statements more readable, but have no effect.</td>
</tr>
<tr>
<td>Modifiers</td>
<td></td>
</tr>
</tbody>
</table>
The BREAK-ON modifier breaks a report into sets of items based on the value of an attribute. A break occurs whenever there is a change in the value of the specified attribute.

**Syntax**

```plaintext
BREAK-ON attrib-name 
"{text...}{options}...text"
```

- **attrib-name**
  - name of the attribute whose detail item values are to be checked for changes; a change in the attribute value causes a break. The attributes are displayed on the report.

- **text**
  - specifies text to be displayed on the break line; can be any string value enclosed in double quotes ("."). The text is justified according to the V/TYP code of the break attribute. If no text is specified, three asterisks (*** are displayed.

- **'options'**
  - can appear anywhere within the text string, and are enclosed in single quotes (') wherever they are used. The valid options are:
    
    - **B** break attribute; inserts the current value of the BREAK-ON attribute into the Ultimate RECALL page heading. This must be used in conjunction with a B option in a HEADING or FOOTING specification. Only one BREAK-ON specification should use a B option (per report).
    
    - **D** break suppress; suppresses the break if only one detail line has been output since the last break on this attribute.
    
    - **L** line suppress; suppresses the blank line preceding the break line. This option is ignored if the U option is also specified.
    
    - **P** page eject; ejects the page after outputting the data associated with this break.
    
    - **R** rollover inhibit; forces all data associated with this control break onto the current page.
    
    - **U** underline: inserts a row of equal signs (=) under all total (subtotal) fields associated with this break.
Modifiers

V value; inserts the current value of the break attribute at the position of the V in the text on the break line. Up to 48 characters can be displayed.

Description

The BREAK-ON modifier causes Ultimate RECALL to check each item for a change in the value of the specified attribute. Correlatives are applied before comparing values, but conversions are not. The values are compared from left to right, character by character (ASCII comparison). Up to a maximum of 48 characters are compared. If the value is the same as the last item output, there is no break and outputting (and totaling) continues. When the value changes, a break occurs.

A break causes Ultimate RECALL to output a break line before continuing with output of the next item (with the new value). The break line has three asterisks (***). The break line also contains subtotals for attributes being totaled (via the TOTAL modifier). All other columns are blank. The break line is preceded and followed by a blank line.

Up to 15 break levels (BREAK-ONs) can be specified; the highest level is the first BREAK-ON in the statement.

For multiple breaks, the break lines are output from the lowest level break to the highest level. The data associated with the lowest level break is printed on the current page (even if the end of the page has been reached). Normal pagination resumes on the second and subsequent data lines unless an option prevents this.
<table>
<thead>
<tr>
<th>Page</th>
<th>Invoice Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>:SORT INVOICE BY VIA BREAK-ON VIA DATE COMPANY</td>
</tr>
<tr>
<td></td>
<td>09:07:24  09 JUL 1992</td>
</tr>
<tr>
<td></td>
<td>INVOICE... Ship Method. Invoice.. Company Name....... Date</td>
</tr>
<tr>
<td>1683</td>
<td>SOP TRUCK 08/09/92 Service Office Products</td>
</tr>
<tr>
<td>1687</td>
<td>SOP TRUCK 06/16/92 Service Office Products</td>
</tr>
<tr>
<td></td>
<td>***</td>
</tr>
<tr>
<td>1681</td>
<td>Truck 08/23/92 Kelley Brothers</td>
</tr>
<tr>
<td>1682</td>
<td>Truck 08/24/92 Kelley Brothers</td>
</tr>
<tr>
<td>1686</td>
<td>Truck 08/21/92 Quality Lighting Products</td>
</tr>
<tr>
<td></td>
<td>***</td>
</tr>
<tr>
<td>1696</td>
<td>UPS 07/01/92 Universal Copiers</td>
</tr>
<tr>
<td></td>
<td>***</td>
</tr>
<tr>
<td></td>
<td>6 items listed.</td>
</tr>
</tbody>
</table>
Modifiers

**COL-HDR-SUPP**
If the report is columnar, the COL-HDR-SUPP modifier suppresses the column heading line that identifies the attribute names in the report and also suppresses the page heading line and end message (same as HDR-SUPP). If the report is non-columnar, the COL-HDR-SUPP modifier suppresses only the page heading and end message; the column headings are not suppressed. If a forms expression is specified, the COL-HDR-SUPP modifier has no effect.

If the COL-HDR-SUPP modifier is specified with the LIST-LABEL or SORT-LABEL command, it also suppresses top-of-form commands.

If the COL-HDR-SUPP modifier is used with S-DUMP or T-DUMP, or with REFORMAT or SREFORMAT where the destination is to tape, it suppresses tape labels.

This is equivalent to the C option in an Ultimate RECALL statement.

```markdown
:SORT INVOICE DATE COMPANY COL-HDR-SUPP
1681 08/23/92 Kelley Brothers
1682 08/24/92 Kelley Brothers
1683 08/09/92 Service Office Products
1686 09/21/92 Quality Lighting Products
1687 08/16/92 Service Office Products
1696 07/01/92 Universal Copiers
```

**DBL-SPC**
The DBL-SPC modifier inserts a blank line (double spaces) between detail lines.

```markdown
:SORT INVOICE DATE COMPANY COL-HDR-SUPP DBL-SPC
1681 08/23/92 Kelley Brothers
1682 08/24/92 Kelley Brothers
1683 08/09/92 Service Office Products
1686 09/21/92 Quality Lighting Products
1687 08/16/92 Service Office Products
1696 07/01/92 Universal Copiers
```
**DET-SUPP**

The DET-SUPP modifier suppresses the display of all detail lines. Only break and total lines are printed on the report.

If BREAK-ON is specified with DET-SUPP, the BREAK-ON value replaces the asterisks on the break line. Up to 48 characters are displayed.

This is equivalent to the D option in an Ultimate RECALL statement.

```plaintext
:SORT INVOICE TOTAL EXT.PRICE DET-SUPP

PAGE 1 09:41:28 17 JUN 1992
INVOICE... Ext Price.
... $917.40
6 items listed.

PAGE 1 09:42:33 17 JUN 1992
  SOX TRUCK $286.80
  Truck     $574.80
  UPS       $55.80
... $917.40
6 items listed.
```
The FOOTING modifier defines a footing for an Ultimate RECALL report.

**Syntax**

```
FOOTING \"{text} {'options'} {text} {'options'}...\"
```

- **text**
  - any string enclosed in double quotes (".

- **'options'**
  - are enclosed in single quotes (') wherever they are used.

The valid options are:

- **B**
  - break; inserts the current value of the BREAK-ON attribute into the footing of each page. This is used when there is a BREAK-ON modifier with the B option specified. Up to 48 characters can be displayed.

- **Bn**
  - break attribute, as above, but inserts the value left-justified in a field of n blanks.

- **I**
  - item-ID; inserts the current item-ID into the footing of each page.

- **In**
  - item-ID, as above, but inserts the item-ID left-justified in a field of n blanks.

- **C**
  - center; centers the current line of the footing using the margins set by the TERM command.

- **Cn**
  - center, as above, but uses a page width of n characters.

- **D**
  - date; inserts the current date at the current position in the footing. The date format is dd mon yyyy.

- **F**
  - filename; inserts the name of the file being reported into the footing.

- **Fn**
  - filename, as above, but inserts the filename left-justified in field of n blanks.

- **L**
  - new line; starts a new line of footing.

- **P**
  - page; inserts the current page number in the footing, right-justified in a field of four blanks.

- **PN**
  - page, as above, but left-justified without blanks.

- **Pn**
  - page, as above, but left-justified in a field of n blanks.
Using Ultimate RECALL

**Description**

The specified footing is output at the bottom of each report page. The footing begins at the left margin of the report (column zero) unless the C option is specified.

A FOOTING modifier can be used anywhere within an Ultimate RECALL statement.

Two contiguous options can be enclosed in one pair of single quotes (for example, 'LC' to begin a new centered line).

```
:SORT INVOICE BY VIA BREAK-ON VIA TOTAL EXT.PRICE COMPANY
FOOTING "'L'Shipping Methods"

PAGE 1  09:38:22  09 JUL 1992

INVOICE... Ship Method. Ext Price. Company Name.......  

1683  SOP TRUCK  $46.20 Service Office Products
       $9.60  
       $30.90  
       $15.40  
       $12.80  
       $20.25  
       $6.10  
       $26.80  
       $35.00  
       $23.20  
       $12.10  

1687  SOP TRUCK  $9.90 Service Office Products
       $9.60  
       $9.90  
       $9.95  
       $9.15  

...  $286.80

Shipping Methods
```
Modifiers

GRAND-TOTAL

The GRAND-TOTAL modifier formats the grand total line and displays specified text, if any. The GRAND-TOTAL modifier does not cause a grand total to be calculated; it only provides a means of including text and formatting characteristics to the grand total line produced by the TOTAL modifier.

Syntax

GRAND-TOTAL "{text...}{'options'}{...text}"

text
options

any string value enclosed in double quotes (").
can appear anywhere within the text string, and are enclosed in single quotes (') wherever they are used.
The valid options are:

L line suppress; suppresses the blank line preceding the grand-total line. This option is ignored if the U option is also specified.
P page eject; ejects the page before printing the grand total line, which can be useful if the grand total is not meaningful to the report (the last page of the report can be discarded).
U underline; inserts a row of equal signs (=) on the blank line between the last detail line and the grand total line under all total fields on the report.

Description

The GRAND-TOTAL modifier can be specified anywhere in the Ultimate RECALL statement. Text specified to be displayed is unconditionally printed left-justified, starting at column 1 (where the item-IDs are displayed on detail lines). The justification code of the item-IDs or an ID-SUPP modifier, if present in the statement, is ignored.
Using Ultimate RECALL

:SORT INVOICE BY VIA BREAK-ON VIA TOTAL EXT_PRICE COMPANY GRAND-TOTAL "Grand Total 'LF'"

PAGE 1  09:46:56  09 JUL 1992

INVOICE... Ship Method. Ext Price. Company Name........
1683 SOP TRUCK $46.20 Service Office Products $9.60

PAGE 5  09:46:59  09 JUL 1992

INVOICE... Ship Method. Ext Price. Company Name........

Grand Total $917.40
6 items listed.

HDR-SUPP

The HDR-SUPP modifier suppresses the page heading line and end message. If used with S-DUMP or T-DUMP, or with REFORMAT or SREFORMAT where the destination is to tape, the HDR-SUPP modifier suppresses the tape label.

This is equivalent to the SUPP modifier and the H option in an Ultimate RECALL statement.

:SORT INVOICE COMPANY HDR-SUPP

INVOICE... Company Name........
1681 Kelley Brothers
1682 Kelley Brothers
1683 Service Office Products
1686 Quality Lighting Products
1687 Service Office Products
1696 Universal Copiers

Ultimate RECALL User's Guide
Confidential and Proprietary to The Ultimate Corp.
HEADING
HEADER

The HEADING modifier defines a heading for an Ultimate RECALL report.

*Note:* HEADER is a synonym for HEADING and can be used in its place.

**Syntax**

```
HEADING \{\{text\} {\'options\'} \{text\} {\'options\'}...\}
```

- **text**
  - any string enclosed in double quotes (".

- **'options'**
  - are enclosed in single quotes (' where they are used.
  - The valid options are:
    - **B** break; inserts the current value of the BREAK-ON attribute into the heading of each page. This is used when there is a BREAK-ON modifier with the B option specified. Up to 48 characters can be displayed.
    - **Bn** break attribute, as above, but inserts the value left-justified in a field of n blanks.
    - **I** item-ID; inserts the current item-ID into the heading of each page.
    - **In** item-ID, as above, but inserts the item-ID left-justified in a field of n blanks.
    - **C** center; centers the current line of the heading using the margins set by the TERM command.
    - **Cn** center, as above, but uses a page width of n characters.
    - **D** date; inserts the current date at the current position in the heading. The date format is dd mon yyyy.
    - **F** filename; inserts the name of the file being reported into the heading.
    - **Fn** filename, as above, but inserts the filename left-justified in field of n blanks.
    - **L** new line; starts a new line of heading.
    - **P** page; inserts the current page number in the heading, right-justified in a field of four blanks.
    - **PN** page, as above, but left-justified with no blanks.
Using Ultimate RECALL

Description

Use of the HEADING modifier suppresses both the default page heading and message line.

A HEADING modifier can be used anywhere within an Ultimate RECALL statement.

If used with S-DUMP or T-DUMP, or with REFORMAT or SREFORMAT where the destination is to tape, the HEADING modifier can be used to specify text for the tape label. The options for the HEADING modifier have no effect with tape.

The specified heading is output at the top of each report page. The heading begins at the left margin of the report (column zero) unless the C option is specified.

Two contiguous options can be enclosed in one pair of single quotes (for example, 'LC' to begin a new centered line).

Pn  page, as above, but left-justified in a field of n blanks.

T   time; inserts the current time and date at the current position in the heading or footing. The time and date format is hh:mm:ss dd mon yyyy.

' '  two consecutive single quote marks insert one single quote mark (') into the heading.
<table>
<thead>
<tr>
<th>PAGE 1</th>
<th>Shipping Method: SOP TRUCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVOICE... Ship Method, Ext Price, Company Name........</td>
<td></td>
</tr>
<tr>
<td>1683</td>
<td>SOP TRUCK</td>
</tr>
<tr>
<td></td>
<td>$46.20 Service Office Products</td>
</tr>
<tr>
<td></td>
<td>$9.60</td>
</tr>
<tr>
<td></td>
<td>$30.90</td>
</tr>
<tr>
<td></td>
<td>$15.40</td>
</tr>
<tr>
<td></td>
<td>$12.80</td>
</tr>
<tr>
<td></td>
<td>$20.25</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PAGE 2</th>
<th>Shipping Method: Truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVOICE... Ship Method, Ext Price, Company Name........</td>
<td></td>
</tr>
<tr>
<td>1681</td>
<td>Truck</td>
</tr>
<tr>
<td></td>
<td>$19.20 Kelley Brothers</td>
</tr>
<tr>
<td></td>
<td>$9.95</td>
</tr>
<tr>
<td></td>
<td>$9.95</td>
</tr>
<tr>
<td></td>
<td>$9.90</td>
</tr>
</tbody>
</table>
The ID-SUPP modifier suppresses the display of item-IDs.

If the ID-SUPP modifier is used with REFORMAT or SREFORMAT, it causes the reformating to skip the first specified attribute and use the second attribute as the item-ID of the new item.

This is equivalent to the I option in an Ultimate RECALL statement.

The LPTR modifier routes the report output to the printer currently assigned to the user's terminal.

This is equivalent to the P option in an Ultimate RECALL statement.

When output is to terminal, the NOPAGE modifier omits the wait at the end of each page of screen output. The entire report is scrolled and displayed without stopping.

If specified with the LIST-LABEL or SORT-LABEL command, the NOPAGE modifier suppresses all headers but the first.

This is equivalent to the N option in an Ultimate RECALL statement.
The ONLY modifier suppresses the use of the default attribute definition items (such as '1' and '2') when no output specifications are given. When the ONLY modifier is specified, just the item-ID column is displayed in the report.

The ONLY modifier must precede the filename.

<table>
<thead>
<tr>
<th>SORT INVOICE</th>
<th>lists all default attributes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAGE 1</td>
<td>12:11:51 09 JUL 1992</td>
</tr>
<tr>
<td>INVOICE : 1681</td>
<td></td>
</tr>
<tr>
<td>Company Name</td>
<td>Kelley Brothers</td>
</tr>
<tr>
<td>Invoice Date</td>
<td>08/23/92</td>
</tr>
<tr>
<td>Description</td>
<td>Diet Cola Regular Cola Diet Root Beer Root Beer Apple Juice Tomato Juice Crackers Water Raspberry Soda Wild Cherry Soda Orange Juice Grapefruit Juice Herb Tea Regular Tea a Cookies</td>
</tr>
<tr>
<td>Qty</td>
<td>1</td>
</tr>
<tr>
<td>Delivery Date</td>
<td>08/20/92</td>
</tr>
<tr>
<td>Ext Price</td>
<td>$19.20</td>
</tr>
<tr>
<td>.</td>
<td></td>
</tr>
<tr>
<td>6 items listed.</td>
<td></td>
</tr>
</tbody>
</table>

| SORT ONLY INVOICE | lists only item-IDs. |
| PAGE 1 | 12:12:09 09 JUL 1992 |
| INVOICE... |
| 1681 |
| 1682 |
| 1683 |
| 1686 |
| 1687 |
| 1696 |
| 6 items listed. |
**SUPP**

The SUPP modifier suppresses the page heading line and end message. If used with S-DUMP or T-DUMP, or with REFORMAT or SREFORMAT where the destination is to tape, the SUPP modifier suppresses the tape label.

This is equivalent to the HDR-SUPP modifier and the H option in an Ultimate RECALL statement.

```
:SORT INVOICE COMPANY SUPP
INVOICE... Company Name........
1681   Kelley Brothers
1682   Kelley Brothers
1683   Service Office Products
1686   Quality Lighting Products
1687   Service Office Products
1696   Universal Copiers
```

**TAPE**

The TAPE modifier causes Ultimate RECALL to retrieve the file and data from a tape file. The file on tape must be in T-DUMP format.

The tape must have been previously attached using the T-ATT command.
TOTAL

The TOTAL modifier accumulates a total value on a report and is used with an attribute name. All attribute totals are reported on a grand total line at the end of the report. If the BREAK-ON modifier is also specified, subtotals are reported on each break line.

Syntax

TOTAL attrib-name {print-limiter}

attrib-name name of the attribute whose detail item values are to be accumulated into the total.

print-limiter specifies limit conditions for the attribute value in order to be output (and totaled).

Description

The TOTAL modifier causes a running total to be accumulated for the specified attribute. Intermediate format values are used; this means that correlatives are applied before the totaling, but conversions are not.

Each value for the specified attribute is added to the total value before it is output. After the last detail line, a total line is output. The total line is identified by three asterisks (***) in the item-ID column, unless the GRAND-TOTAL or ID-SUPP modifier has been specified. If GRAND-TOTAL is specified with text, the text is output instead of the asterisks. If ID-SUPP is specified, the asterisks are not displayed.

If a report does not contain any BREAK-ON modifiers, there is only one total line (the grand-total line).

Totals are accumulated after correlatives are applied, but before conversions are. In accumulating totals, Ultimate RECALL stops at the first non-numeric character. This means that when you use an attribute definition item for a total, be sure it does not have a correlative that formats the value with non-numeric characters. If it does, the total will not be accurate. (For more information on correlatives and conversions, see Chapter 5.)

All detail lines can be suppressed from a report that contains totals if the statement contains the DET-SUPP modifier. Only break lines and total lines are printed.
### INVOICE

**Ext Price, Company Name**

<table>
<thead>
<tr>
<th>Company Name</th>
<th>Ext Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kelley Brothers</td>
<td>$353.85</td>
</tr>
<tr>
<td>Quality Lighting Products</td>
<td>$220.95</td>
</tr>
<tr>
<td>Service Office Products</td>
<td>$286.80</td>
</tr>
<tr>
<td>Universal Copiers</td>
<td>$555.80</td>
</tr>
<tr>
<td></td>
<td>$917.40</td>
</tr>
</tbody>
</table>

6 items listed.
**USING**

The USING modifier selects and formats items in one file using attribute definition items from another file.

**Syntax**

```
USING {DICT} filename
```

- **DICT** references the dictionary portion of the file.
- **filename** name of the file in which to find the definition items to use to produce and format the report.

**Description**

Only one USING modifier can be used per Ultimate RECALL statement.

Only attribute definition items are used, not data.

A USING phrase can be placed anywhere in the Ultimate RECALL statement. It can precede or follow selection criteria, output specifications, or any other parameters.

The USING modifier can be used, for example, during program development to set up a test dictionary while creating new attribute definition items. It can also be used to specify an existing dictionary to report on new, temporary, or test data files without having to build a new dictionary.

In a production environment, the USING modifier may be effective in controlling access to specific data in a file. Instead of having all definition items in a single dictionary, sub-dictionaries could be created (shared dictionary format: `dictname,dataname`) with the definitions appropriate to certain users.

```
:LIST USING DICT CUSTOMERS TEST
```

<table>
<thead>
<tr>
<th>PAGE</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST</td>
<td>1</td>
</tr>
<tr>
<td>Company</td>
<td>Service Office Products</td>
</tr>
<tr>
<td>Contact</td>
<td>Lee</td>
</tr>
<tr>
<td>Address</td>
<td>4512 Orange</td>
</tr>
<tr>
<td>City</td>
<td>Santa Ana, CA 92222</td>
</tr>
<tr>
<td>Phone Number</td>
<td>714/555-1234</td>
</tr>
</tbody>
</table>

End of list
WITHIN

Some files are designed to handle items that have an exploding tree structure, such as bill-of-material files. A data file item-ID can identify a unit that is made up of multiple components; each component is also an item with a separate item-ID in the file. The WITHIN modifier allows reporting on the product structure of a specified exploding item in this type of file.

Syntax

```
RECALL.cmd WITHIN filename 'item-ID' {options...}
```

- **RECALL.cmd** can be LIST, COUNT, STAT, or SUM. For information on these commands, see Chapter 3.
- **filename** identifies a file with an exploding tree structure. This file must have a V correlative in its file definition item.
- **item-ID** specifies the primary item to retrieve and explode.
- **options** any valid output specifications, modifiers, and options for the specified command.

Description

In order to use the WITHIN modifier, the file must have a V correlative defined in its file definition item. The V correlative defines the attribute to explode in order to build the report. For information on the V processing code, see Chapter 5.

The WITHIN connective works with the V correlative to extract all item-IDs that are in the attribute specified in the V correlative. The first item must be listed in the Ultimate RECALL statement. If additional items are specified, they are ignored. The items pointed to by the first item can themselves point to additional items. Up to 20 levels can be nested.

Ultimate RECALL generates a special column called LEVELS as the first column of the report and the item-ID column moves to the right. The LEVELS column reports the levels (indented explosion) that comprise the specified item-ID. The specified item-ID is level 1; the items it directly references are level 2; the items referenced by level 2 are level 3, and so on.

By using the WITHIN connective, all items that relate to the finished product can be reported on.
:LIST WITHIN ASSEMBLY 'A100'

LEVEL ASSEMBLY.. DESCRIPTION........ SUB-ASSEMBLY...

1 A100 Finished product A10
    A11
    A12
2 A10 Component A
2 A11 Component B
    A13
    A14
3 A13 Raw material for B
3 A14 Raw material for B
2 A12 Component C

6 items listed.
Using Ultimate RECALL

**Throwaway Modifiers**

An Ultimate RECALL statement can include a number of modifiers that have no effect on the report itself. These *throwaway* modifiers are included in the standard Ultimate RECALL vocabulary to enhance the English-like syntax of Ultimate RECALL statements.

The following are standard throwaway modifiers:

- A
- AN
- ANY
- ARE
- DATA
- FILE
- FOR
- IN
- ITEMS
- OF
- PAGE
- THE

These modifiers can be placed anywhere in the statement and have no effect. The following two statements are equivalent:

```
:SORT THE INVOICE FILE FOR ITEMS AFTER '1682'
```

```
:SORT INVOICE > '1682'
```
Options

If options are used, they must be placed at the end of the Ultimate RECALL statement and enclosed in parentheses.

The options that are available for Ultimate RECALL depend on the command. Table 2-4 lists the options used by Ultimate RECALL.
### Table 2-4. Options (1 of 3)

<table>
<thead>
<tr>
<th>Option</th>
<th>Commands that Use Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>LIST SORT</td>
<td>allows forms alignment. This option is used by forms only.</td>
</tr>
<tr>
<td>B</td>
<td>LIST SORT</td>
<td>prints background item. This option is used by forms only.</td>
</tr>
<tr>
<td>C</td>
<td>LIST SORT</td>
<td>if the report is columnar, suppresses the page heading line and end message and also omits the column heading line that identifies the attribute names in the report. If the report is non-columnar, suppresses only the page heading and end message; the column headings are not suppressed. (This is equivalent to the COL-HDR-SUPP modifier.)</td>
</tr>
<tr>
<td>C</td>
<td>LIST-LABEL SORT-LABEL</td>
<td>suppresses the page heading line and top-of-form commands. (This is equivalent to the COL-HDR-SUPP modifier.)</td>
</tr>
<tr>
<td>C</td>
<td>REFORMAT SREFORMAT</td>
<td>if the destination of the reformatted items is tape, suppresses the tape label. (This is equivalent to the COL-HDR-SUPP modifier.)</td>
</tr>
<tr>
<td>D</td>
<td>LIST LIST-LABEL SORT SORT-LABEL</td>
<td>suppresses all detail lines on report. Any break lines and totals specified in the Ultimate RECALL command are output. (This is equivalent to the DET-SUPP modifier.)</td>
</tr>
<tr>
<td>F</td>
<td>LIST-ITEM SORT-ITEM</td>
<td>forces a new page after every item.</td>
</tr>
<tr>
<td>H</td>
<td>LIST LIST-LABEL REFORMAT SORT SORT-LABEL SREFORMAT</td>
<td>suppresses the page heading line and the &quot;n items listed&quot; message at the end of the report. (This is equivalent to the HDR-SUPP or SUPP modifiers.)</td>
</tr>
</tbody>
</table>
Table 2-4. Options (2 of 3)

<table>
<thead>
<tr>
<th>Option</th>
<th>Commands that Use Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>S-DUMP, T-DUMP</td>
<td>suppresses the tape label. (This is equivalent to the HDR-SUPP or SUPP modifiers.)</td>
</tr>
<tr>
<td>I</td>
<td>LIST, LIST-ITEM, LIST-LABEL, S-DUMP, SORT, SORT-ITEM, SORT-LABEL, T-DUMP</td>
<td>suppresses item-ID listing. (This is equivalent to the ID-SUPP modifier.)</td>
</tr>
<tr>
<td>I</td>
<td>REFORMAT, SREFORMAT</td>
<td>causes the reformatting to skip the first specified attribute and use the second attribute as the item-ID of the new item. (This is equivalent to the ID-SUPP modifier.)</td>
</tr>
<tr>
<td>M</td>
<td>LIST, SORT</td>
<td>lists multiple items per page. This option is used by forms only.</td>
</tr>
<tr>
<td>N</td>
<td>All</td>
<td>suppresses wait at end of page. (This is equivalent to the NOPAGE modifier.)</td>
</tr>
<tr>
<td>O</td>
<td>T-LOAD</td>
<td>overwrites the item in the file if the item on tape has same item-ID.</td>
</tr>
<tr>
<td>P</td>
<td>All</td>
<td>routes output to the spooler. (This is equivalent to the LPTR modifier.)</td>
</tr>
<tr>
<td>S</td>
<td>HASH-TEST, ISTAT</td>
<td>suppresses the histogram and shows only summary statistics.</td>
</tr>
<tr>
<td>S</td>
<td>LIST-ITEM, SORT-ITEM</td>
<td>suppresses display of line numbers.</td>
</tr>
</tbody>
</table>
### Table 2-4. Options (3 of 3)

<table>
<thead>
<tr>
<th>Option</th>
<th>Commands that Use Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>W</strong></td>
<td>LIST, LIST-LABEL, REFORMAT, SELECT, SORT, SORT-LABEL, SREFORMAT, SSELECT</td>
<td>allows BASIC subroutine to write to files it opens.</td>
</tr>
<tr>
<td><strong>X</strong></td>
<td>LIST-ITEM, SORT-ITEM</td>
<td>displays output in hexadecimal.</td>
</tr>
<tr>
<td><strong>Z</strong></td>
<td>LIST, SORT</td>
<td>resets page number to 1 for each form. This option is used by forms only.</td>
</tr>
</tbody>
</table>
3 Ultimate RECALL Commands

Ultimate RECALL is accessed through a set of system commands. (An Ultimate system command is any command that can be specified from the TCL level or used anywhere that a TCL level command is valid.)

Table 3-1 lists the Ultimate RECALL system commands. These commands are discussed in alphabetical order in this section.

For information on the general elements in an Ultimate RECALL statement, see Chapter 2, Forming an Ultimate RECALL Statement.
### Table 3-1. Ultimate RECALL Commands (1 of 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECK-SUM</td>
<td>generates a checksum of selected items.</td>
</tr>
<tr>
<td>COUNT</td>
<td>outputs the total number of selected items.</td>
</tr>
<tr>
<td>HASH-TEST</td>
<td>produces statistics showing the distribution of items based on a test modulo.</td>
</tr>
<tr>
<td>ISTAT</td>
<td>produces statistics showing the distribution of items based on the current modulo.</td>
</tr>
<tr>
<td>LIST</td>
<td>outputs selected items and attributes in a formatted report, based on the attribute definition items.</td>
</tr>
<tr>
<td>LIST-ITEM</td>
<td>lists contents of selected items.</td>
</tr>
<tr>
<td>LIST-LABEL</td>
<td>outputs selected items and attributes in a label format; prompts for parameters such as number of items to print across the report page.</td>
</tr>
<tr>
<td>REFORMAT</td>
<td>produces a new file from selected items and attributes, on disk or tape.</td>
</tr>
<tr>
<td>S-DUMP</td>
<td>outputs selected and sorted file items to tape.</td>
</tr>
<tr>
<td>SELECT</td>
<td>produces a select-list that contains selected item-ids and, optionally, specified attributes.</td>
</tr>
<tr>
<td>SORT</td>
<td>outputs selected items and attributes, sorted by specified attribute values, in a formatted report, based on the attribute definition items.</td>
</tr>
<tr>
<td>SORT-ITEM</td>
<td>lists contents of selected items, sorted by specified attribute values.</td>
</tr>
<tr>
<td>SORT-LABEL</td>
<td>outputs selected items and attributes, sorted by specified attributes, in a label format; prompts for parameters such as number of items to print across the report page.</td>
</tr>
</tbody>
</table>
# Table 3-1. Ultimate RECALL Commands (2 of 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SREFORMAT</td>
<td>produces a new file from selected items and attributes, sorted by specified attribute values, on disk or tape.</td>
</tr>
<tr>
<td>SSELECT</td>
<td>produces a select-list that contains selected item-IDs and, optionally, specified attributes, sorted by specified attribute values.</td>
</tr>
<tr>
<td>STAT</td>
<td>outputs the total value of a specified attribute in all selected items; also outputs the total number of items selected and the average value of the attribute per item.</td>
</tr>
<tr>
<td>SUM</td>
<td>outputs the total value of a specified attribute in all selected items.</td>
</tr>
<tr>
<td>T-DUMP</td>
<td>outputs selected items to tape.</td>
</tr>
<tr>
<td>T-LOAD</td>
<td>loads selected or specified file items from the attached tape.</td>
</tr>
</tbody>
</table>
CHECK-SUM

The CHECK-SUM command generates a checksum for file items.

**Syntax**

```
CHECK-SUM filename {itemlist} {sel-criteria} {attrib-name} {{(P}
```

- `filename`: specifies file to be checksummed.
- `itemlist`: specifies items to checksum; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.
- `sel-criteria`: conditions that must be met in order for the item to be included in the checksum calculations. If no selection criteria are present, all items being considered are selected.
- `attrib-name`: name of attribute to be checksummed; if the attribute number of `attrib-name` is 9999, the entire item is checksummed.
- `(P)`: routes output to the spooler. (This is equivalent to the LPTR modifier.)

**Description**

The checksum is a calculation based on the binary value of each character times a positional value. This yields a checksum that has a high probability of being unique for a given character string.

To use checksums, you should issue CHECK-SUM commands for all files, or portions of files, to be verified; the output statistics should be kept. Subsequently, the CHECK-SUM command can be re-issued to verify that the checksum statistics have not changed. The checksum must be recalculated whenever you update the file.

The checksum statistics are displayed in the following form:

```
Byte statistics for: name:
  Total = t  Average = a  Items = i  Cksum = c  Bits = b
```
where
name  attribute name, if specified; otherwise filename.
t  total number of bytes in the attribute or items.
a  average number of bytes.
i  number of items.
c  checksum.
b  bit count.

Attribute marks are included in the statistics. When an entire item is processed, the count field, item-ID, and first attribute mark are included in the calculations.

<table>
<thead>
<tr>
<th>CHECK-SUM</th>
<th>INVOICE</th>
<th>Checksums the entire file.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte statistics for : INVOICE</td>
<td>Total = 1226  Average = 204.33  Items = 6  Cksum = 14748788  Bits = 5183</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHECK-SUM</th>
<th>INVOICE NAME</th>
<th>Checksums the attribute NAME.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte statistics for : Contact</td>
<td>Total = 34  Average = 5.66  Items = 6  Cksum = 17601  Bits = 150</td>
<td></td>
</tr>
</tbody>
</table>
Ultimate RECALL Commands

COUNT

The COUNT command counts the number of items that meet the conditions specified by the itemlist and selection criteria. The output of the command is a report of items counted.

Syntax

COUNT filename \{itemlist\} \{sel-criteria\} \{(P\}

filename specifies file to be counted.

itemlist specifies items to be counted; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.

sel-criteria conditions that must be met in order for the item to be included in the count. If no selection criteria are present, all items being considered are selected.

(P routes output to the spooler. (This is equivalent to the LPTR modifier.)

Description

The COUNT command uses the specified file, items, and selection criteria to determine a count of items that meet the conditions.

The count is displayed in the following form:

\[ n \text{ items counted}. \]

where \( n \) is the number of items that meet the command specifications.

The maximum number of items that can be counted is 2,147,483,647 (which has the hexadecimal value \( \text{X'7FFFFFFF} \)).

<table>
<thead>
<tr>
<th>:COUNT INVOICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 items counted.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>:COUNT INVOICE WITH NAME LIKE &quot;JERRY&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 items counted.</td>
</tr>
</tbody>
</table>
The HASH-TEST command uses a test modulo to provide file management information about a file. The file must currently exist.

**Syntax**

```
HASH-TEST filename {itemlist} {sel-criteria} {(options}
```

- **filename** specifies file to be hash-tested.
- **itemlist** specifies items to test; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.
- **sel-criteria** conditions that must be met in order for the item to be included in the hash-test. If no selection criteria are present, all items being considered are selected.
- **(options** the following options are available:
  - **N** suppresses wait at end of page. (This is equivalent to the NOPAGE modifier.)
  - **P** routes output to the spooler. (This is equivalent to the LPTR modifier.)
  - **S** suppresses display of histogram.

**Description**

The HASH-TEST command shows how items in a file would hash into groups, given a test modulo.

HASH-TEST can be used to determine the best modulo for a given file. It is typically used prior to file reallocation. It also gives the same item and byte count statistics as the ISTAT command.

If used without the **S** option, HASH-TEST displays a file hashing histogram, which shows graphically the number of items that hash into each group.

When HASH-TEST is invoked, it first prompts for the modulo:

```
Test modulo/<CR>:
```

 HASH-TEST
To use the current modulo, press RETURN. To use a different modulo, enter the modulo (number of groups) to use. HASH-TEST then prompts for the hashing algorithm:

Algorithm (1/2/<CR>):

To use the current hashing algorithm, press RETURN. To use the standard algorithm, enter 1. To use the CRC algorithm, enter 2.

(For more information on hashing algorithms and modulos, see the Ultimate System Management Guide.)

:HASH-TEST INVOICE

Test modulo/<CR>: 3
Algorithm (1/2/<CR>):

FILE= INVOICE MODULO= 3 SEPAR= 1 ALGORITHM= 1 09:58:39 09 JUL 1992
FRAMES  BYTES ITEMS
 2  506  2 "»>
 2  557  3 "»»>
 1  420  2 "»»
 5

Item count= 7, byte count= 1483, avg. bytes/item= 211.8
avg. items/group= 2.3, std. deviation= .5, avg. bytes/group= 494.3.
The ISTAT command provides file management information about an existing file.

Syntax

```
ISTAT filename {itemlist} {sel-criteria} {(options}
```

- **filename**: specifies file to be used to produce statistics.
- **itemlist**: specifies items to use to calculate statistics; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.
- **sel-criteria**: conditions that must be met in order for the item to be included in the statistics. If no selection criteria are present, all items being considered are selected.
- **(options****: the following options are available:
  - **N**: suppresses wait at end of page. (This is equivalent to the NOPAGE modifier.)
  - **P**: routes output to the spooler. (This is equivalent to the LPTR modifier.)
  - **S**: suppresses display of histogram.

Description

The ISTAT command provides a file hashing histogram (bar graph) for the selected items in the file (if the S option is not used), as well as statistics on items, item-IDs, groups, and frames. The histogram shows graphically the number of items that hash into each group.

The size of an item includes the number of bytes in the item-ID and the item body, plus one for the trailing segment mark. If an item is extended (extended items are items over 32Kb), the pointer size is counted, but neither the item body size nor the frames used for extended items are counted.

The histogram displays the following information:

- **FRAMES**: number of frames in each group, followed by the total number of frames in the file.
- **BYTES**: number of bytes in each group.
Ultimate RECALL Commands

ITEMS

number of items in each group.

The statistical report displays the following information:

Item count

total number of items in the file.

byte count

total number of bytes in the file.

avg bytes/item

average number of bytes in each item
(byte count divided by item count, rounded).

avg items/group

average number of items in each group
(item count divided by group count, rounded).

std deviation

standard deviation of items per group.

avg bytes/group

average number of bytes in each group
(byte count divided by group count, rounded).

:ISTAT INVOICE (S

FILE= :VOICE MODULO= 3 SEPAR= 1 ALGORITHM= 1 10:12:37 09 JUL 1992
FRAMES BYTES ITEMS
5

Item count= 6, byte count= 1232, avg. bytes/item= 205.3
avg. items/group= 2.0, std. deviation= 1.0, avg. bytes/group= 410.6.
LIST

The LIST command generates formatted output of selected items and attributes.

Syntax

```
LIST filename {itemlist} {sel-criteria} {output-
specifications} {print-limiters} {modifiers} {options}
```

filename
name of file to be listed.

itemlist
specifies items to list; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.

sel-criteria
provides the tests that the items chosen for consideration must pass in order to be included in the list. If no selection criteria are present, all items being considered are selected.

output-specifications
indicates which attributes are to be listed; these determine which fields of information are included on the report, and how they are formatted.

print-limiters
indicates criteria that a value in the specified attribute must meet in order to be included on the report. If no print-limiters are present, all the values of a multivalued attribute are included.

modifiers
control the format and layout of the report. Any modifier described in Chapter 2 can be included.

(options
the following options are available:

A alignment; forms only.

B background; forms only.

C if the report is columnar, suppresses the page heading line and end message and also omits the column heading line that identifies the attribute names in the report. If the report is non-columnar,
### Description

The LIST command displays or prints selected data from selected items in the order in which they are specified in the itemlist. If no itemlist is specified, they are output in the order in which the groups and items are stored.

If forms expressions are used, output is formatted according to the forms expressions. If no forms expressions are present, a columnar format is used if the number of attributes and their names being listed fit in the output page width, otherwise a non-columnar format is used. (Columnar and non-columnar formats are described in Chapter 1. Forms expressions are described in Chapter 6.)

For information on producing a sorted report, see SORT, described alphabetically in this chapter.

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>suppresses only the page heading and end message; the column headings are not suppressed. (This is equivalent to the COL-HDR-SUPP modifier.)</td>
</tr>
<tr>
<td>12</td>
<td>suppresses all detail lines on report. Any break lines and totals specified in the Ultimate RECALL command are output. (This is equivalent to the DET-SUPP modifier.)</td>
</tr>
<tr>
<td>H</td>
<td>suppresses the page heading line (time and date on the left, page number on the right) and the &quot;n items listed&quot; message at the end of the report. (This is equivalent to the HDR-SUPP or SUPP modifiers.)</td>
</tr>
<tr>
<td>I</td>
<td>suppresses item-ID listing. (This is equivalent to the ID-SUPP modifier.)</td>
</tr>
<tr>
<td>M</td>
<td>specifies multiple items per page; forms only.</td>
</tr>
<tr>
<td>N</td>
<td>suppresses wait at end of page. (This is equivalent to the NOPAGE modifier.)</td>
</tr>
<tr>
<td>P</td>
<td>routes output to the spooler. (This is equivalent to the LPTR modifier.)</td>
</tr>
<tr>
<td>W</td>
<td>allows BASIC subroutine to write to files it opens.</td>
</tr>
<tr>
<td>Z</td>
<td>resets page number to 1 for each form; forms only.</td>
</tr>
<tr>
<td>Description</td>
<td>Qty</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>Diet Cola</td>
<td>2</td>
</tr>
<tr>
<td>Regular Cola</td>
<td>1</td>
</tr>
<tr>
<td>Diet Root Beer</td>
<td>1</td>
</tr>
<tr>
<td>Root Beer</td>
<td>1</td>
</tr>
<tr>
<td>Apple Juice</td>
<td>2</td>
</tr>
<tr>
<td>Tomato Juice</td>
<td>1</td>
</tr>
<tr>
<td>Crackers</td>
<td>1</td>
</tr>
<tr>
<td>Water</td>
<td>1</td>
</tr>
<tr>
<td>Raspberry Soda</td>
<td>1</td>
</tr>
<tr>
<td>Wild Cherry Soda</td>
<td>1</td>
</tr>
<tr>
<td>Orange Juice</td>
<td>4</td>
</tr>
<tr>
<td>Grapefruit Juice</td>
<td>1</td>
</tr>
<tr>
<td>Herb Tea</td>
<td>1</td>
</tr>
<tr>
<td>Regular Tea</td>
<td>1</td>
</tr>
<tr>
<td>Cookies</td>
<td>6</td>
</tr>
<tr>
<td>Diet Cola</td>
<td>2</td>
</tr>
<tr>
<td>Orange Juice</td>
<td>1</td>
</tr>
<tr>
<td>Cranberry Juice</td>
<td>1</td>
</tr>
<tr>
<td>Grapefruit Juice</td>
<td>1</td>
</tr>
<tr>
<td>Hot Chocolate Regular</td>
<td>1</td>
</tr>
<tr>
<td>Hot Chocolate Sugar Free</td>
<td>3</td>
</tr>
</tbody>
</table>
LIST-ITEM

The LIST-ITEM command lists the attributes of selected items.

Syntax

LIST-ITEM  filename  {itemlist}  {sel-criteria}  {modifiers}  {{options}}

filename  name of file to be listed.

itemlist  specifies items to list; item-IDs should be enclosed in single quotes (''). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.

sel-criteria  provide the tests that the items chosen for consideration must pass in order to be included in the list. If no selection criteria are present, all items being considered are selected.

modifiers  control the format and layout of the output. The following modifiers described in Chapter 2 are available:

  DBL-SPC  doublespaces output.
  FOOTING  displays specified footing.
  HDR-SUPP  suppresses the page heading line. (This is equivalent to the H option.)
  HEADING  displays specified heading.
  ID-SUPP  suppresses item-ID listing. (This is equivalent to the I option.)
  LPTR  routes output to spooler. (This is equivalent to the P option.)
  NOPAGE  no end-of-page waiting. (This is equivalent to the N option.)
  SUPP  suppresses the page heading line. (This is equivalent to the H option.)
  TAPE  obtains items from tape.
  WITHIN  lists exploding tree structure.
The LIST-ITEM command is similar to the CT command, but it provides many of the Ultimate RECALL capabilities, such as selection criteria and headings or other output formatting.

The entire contents of the selected items are displayed in the order in which they are specified in the itemlist. If no itemlist is specified, they are output in the order in which the groups and items are stored. Attribute numbers are displayed at the left margin.

No correlative or conversion code processing takes place.

For information on producing a sorted report, see SORT-ITEM, described alphabetically in this chapter.
: LIST-ITEM INVOICE '1682'

PAGE 1

1682
001 Kelley Brothers
002 9003
003 7055|6032|6065|6068|5011|5015
004 2|1|1|1|1|3
005 8992|8997|8997|8997|8997|8997
006 O|O|O|O|O|O
007 Jerry
008 12345 Main Street
009 92006
010 Net 30
011 Truck
012 7145556789

17:27:19 10 MAR 1992
LIST-LABEL

The LIST-LABEL command generates formatted output of data. Item data can be grouped into blocks, with several blocks placed across the page as in a set of mailing labels.

Syntax

LIST-LABEL filename {itemlist} {sel-criteria} {output-specifications} {print-limiters} {modifiers} {options}

filename name of file to be listed.

itemlist specifies items to list; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.

sel-criteria provides the tests that the items chosen for consideration must pass in order to be included in the list. If no selection criteria are present, all items being considered are selected.

output-specifications indicates which attributes are to be listed; these determine which fields of information are included on the report, and how they are formatted.

print-limiters indicates criteria that a value in the specified attribute must meet in order to be included on the report. If no print-limiters are present, all the values will be included for a multivalued attribute.

modifiers controls the format and layout of the output. Any modifier described in Chapter 2, except WITHIN, can be included.

(options the following options are available:

C suppresses the page heading line and end message; also suppresses top-of-forms.

D suppresses all detail lines on report. Any break lines and totals specified in the Ultimate RECALL command are output. (This is equivalent to the DET-SUPP modifier.)
Ultimate RECALL Commands

H suppresses the page heading line (time and date on the left, page number on the right) and the "n items listed" message at the end of the report. (This is equivalent to the HDR-SUPP or SUPP modifiers.)

I suppresses item-ID listing. (This is equivalent to the ID-SUPP modifier.)

N suppresses wait at end of page. LIST-LABEL output is generated as one continuous report page; all headers but the first is suppressed. (This is equivalent to the NOPAGE modifier.)

P routes output to the spooler. (This is equivalent to the LPTR modifier.)

W allows BASIC subroutine to write to files it opens.

Description

Before searching the file and creating the list, the LIST-LABEL command displays a question mark (?) to prompt for an additional set of parameters. You must enter the information needed to create the desired arrangement of attributes and blocks per line in the following order:

count,rows,skip,indent,size,space{,|{|S}}

where

- count number of items (labels) across the page.
- rows number of lines printed per label (height of each label, in rows).
- skip number of lines to skip between each label (vertical spacing between labels, in rows).
- indent number of spaces to indent from the left margin (to allow for printing specified text).
- size maximum number of characters to be printed in an attribute (label width, in columns).
- space number of spaces between items (horizontal spacing between labels, in columns).
- C specifies that null attributes are not to be printed (if omitted, null values are printed as all blanks).
- S specifies that a new line of labels is to start after each control break (used with the BREAK-ON connective).
LIST-LABEL

The values used must conform to the range:

\[(\text{count} * (\text{size} + \text{space}) + \text{indent}) \leq \text{page width}\]

where page width is the number defined in the TERM command for the current output device (printer or terminal). If the maximum number of characters specified exceeds the page width, the system displays a message similar to the following, where \(n\) is the invalid parameter:

[290] The range of the parameter "\(n\)" is not acceptable

If indent is non-zero, the command displays a question mark (?) to prompt for the text to be printed. One question mark is displayed for each row specified for the parameter rows. When the listing is printed, this text is displayed in the indent area. (To specify null text for a line, press RETURN at the corresponding prompt.)

If indent is zero or null, no text is displayed at the left margin; instead the first block of data begins at column 1.

The standard heading is displayed at the top of each page, unless suppressed by the COL-HDR-SUPP or HDR-SUPP modifier, or the C or H option.

If headings are suppressed by the COL-HDR-SUPP modifier or the C option, pagination and all top-of-forms are also suppressed. This produces a continuous forms format without page breaks.

The LIST-LABEL command ignores VTYP and VMAX. All data is printed left justified on one line using as many characters as specified in the size parameter.

The items are output in the order in which they are specified in the itemlist. If no itemlist is specified, they are output in the order in which the groups and items are stored. (For information on producing sorted labels, see SORT-LABEL, described alphabetically in this chapter.)
Ultimate RECALL Commands

:LIST-LABEL CUSTOMERS (CI produces a continuous forms format without page breaks.

<table>
<thead>
<tr>
<th>Quality Lighting Products</th>
<th>Service Office Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mat</td>
<td>Lee</td>
</tr>
<tr>
<td>5454 W. Mariposa</td>
<td>4512 Orange</td>
</tr>
<tr>
<td>Downey, CA 91710</td>
<td>Santa Ana, CA 92222</td>
</tr>
</tbody>
</table>

| Universal Copiers            | Kelley Brothers         |
| Marina                       | Jerry                   |
| 211 Westgate                 | 12345 Main Street       |
| Long Beach, CA 91832         | Anaheim, CA 92006       |

| Service Office Products      |                           |
| Shelby                       |                           |
| 3114 Paradise Drive          |                           |
| San Fernando, CA 91340       |                           |
The REFORMAT command creates new items by reformatting existing items. The new items can be placed in a separate file, in the current file, or on tape.

**Syntax**

```
REFORMAT filename {itemlist} {sel-criteria} {output-specifications} {print-limiters} {modifiers} {options}
```

- **filename**: name of file that contains items to be reformatted.
- **itemlist**: specifies items to reformat; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.
- **sel-criteria**: provides the tests that the items chosen for consideration must pass in order to be reformatted. If no selection criteria are present, all items being considered are selected.
- **output-specifications**: indicates the attributes that are to be used in the new item; if not specified and there are no default output specifications, no items are created.
- **print-limiters**: indicates criteria that a value in the specified attribute must meet in order to be reformatted. If no print-limiters are present, all the values are included for a multivalued attribute.
- **modifiers**: control the format and layout of the output. The following modifiers described in Chapter 2 are available:
  - **COL-HDR-SUPP**: suppresses the tape label; meaningful only if destination is tape. (This is equivalent to the C option.)
  - **HDR-SUPP**: suppresses the tape label; meaningful only if destination is tape. (This is equivalent to the H option.)
  - **HEADING "text"**: specifies text to be included in the output.
Ultimate RECALL Commands

Description

When the REFORMAT command is invoked, it prompts for the destination file name:

```
File name:
```

- to store the reformatted items in a different file on disk, enter the filename.
- to write the items to tape, enter the word TAPE.
- to store the reformatted items back into the current file, press RETURN.

The following options are available:

- **C** suppresses the tape label; meaningful only if destination is tape. (This is equivalent to the COL-HDR-SUPP modifier.)
- **H** suppresses the tape label; meaningful only if destination is tape. (This is equivalent to the HDR-SUPP or SUPP modifiers.)
- **I** suppresses item-ID; causes the reformatting to skip the first specified attribute and use the second attribute as the item-ID of the new item. (This is equivalent to the ID-SUPP modifier.)
- **P** when the destination is to tape, routes the list of item-IDs to the spooler; meaningful only if destination is tape. (This is equivalent to the LPTR modifier.)
- **W** allows BASIC subroutine to write to files it opens.

**ID-SUPP** suppresses item-ID; causes the reformatting to skip the first specified attribute and use the second attribute as the item-ID of the new item. (This is equivalent to the I option.)

**LPTR** when the destination is to tape, routes the list of item-IDs to the spooler; meaningful only if destination is tape. (This is equivalent to the P option.)

**TAPE** gets data from file on tape.
Note: When reformatting to the current file, you should specify an itemlist or use a select-list; otherwise, the command may execute an infinite loop.

The value of the first attribute defined by the output specifications is used as the item-ID. The remaining attributes make up the item. The items are reformatted in the order in which they are specified in the itemlist. If no itemlist is specified, they are reformatted in the order in which the groups and items are stored. (For information on producing sorted output, see SREFORMAT, described alphabetically in this chapter.)

Reformatting to Tape

When the reformatting is to tape, a tape label containing the filename, tape block (record) length in hexadecimal, and the current time and date is written at the start of the dump, before any items. You can suppress the heading by specifying the HDR-SUPP modifier or the H option. You can specify additional information for the header by using the HEADING modifier.

Before executing the command, you should issue a T-ATT. Specify the blocksize equal to the maximum length of the reformatted items. As each item is reformatted, the specified attributes for the new item are concatenated, and either truncated or padded at the end with nulls (X'00') to obtain a block the same length as the blocksize specified by the T-ATT command.

One tape block is written for each item. Item-IDs from the file being reformatted are displayed as the items are written to tape unless the ID-SUPP (I option) is specified. After all the data has been written, two EOFs are written to terminate the tape.

Reformatting to tape is intended primarily to create tapes to be used with non-PICK® systems. Each attribute should be formatted to a fixed length and the tape should be attached at a blocksize equal to the sum of the lengths of the specified attributes.
**Ultimate RECALL Commands**

\[ \text{:REFORMAT INVOICE COMPANY NAME ADDRESS CITY STATE ZIP} \quad \text{*A12} \]

- **File name:** TEMP
  - Creates a file with name as the item-ID, and with company name, city, state, zip, and phone number (*A12) attributes.

\[ \text{:REFORMAT TEMP \quad \text{*A9998} \quad \text{*A1 \quad *A0 \quad *A2 \quad *A3 \quad *A4 \quad *A5 \quad *A6}} \]

- **File name:** CUSTOMERS
  - Creates a file with a sequential number as the item-ID; the item-ID in the TEMP file is attribute 2, followed by the remaining attributes.

\[ \text{:T-ATT 0 100} \]

- **Tape 0 attached**
  - **Block size:** 100

\[ \text{:REFORMAT CUSTOMERS COMPANY NAME ADDRESS CITY STATE ZIP (E}} \]

- **File name:** TAPE
  - **Block size:** 100
    - 1 1
    - 2 2
    - 3 3
    - 4 4
    - 5 5
  - 5 items dumped.

\[ \text{:T-REW} \]

- **Block size:** 100
  - Rewinds the tape to display the tape records.

\[ \text{:T-READ} \]

- **Block size:** 100
  - Displays the tape records.

End of file
S-DUMP

The S-DUMP command copies the contents of a specified file to tape in a sorted sequence.

Syntax

S-DUMP filename \{itemlist\} \{sel-criteria\} \{sort-criteria\} \{modifiers\} \{options\}

filename  name of file to be sorted and dumped.
itemlist  specifies items to dump; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.

sel-criteria  provides the tests that the items chosen for consideration must pass in order to be included in the dump. If no selection criteria are present, all items being considered are selected.

sort-criteria  specifies attributes to use as sort keys to resequence the items for the report output. A number of sort criteria can be used in order to create a multi-level sorted report. The sort can be in ascending or descending order. If no sort criteria are present, the items are sorted by item-ID in ascending order.

modifiers  control the format and layout of the output. The following modifiers described in Chapter 2 are available:

HDR-SUPP  suppresses the tape label. (This is equivalent to the H option.)

HEADING "text"  specifies text to be included in the tape label.

ID-SUPP  suppresses item-ID listing. (This is equivalent to the I option.)

LPTR  routes output to spooler. (This is equivalent to the P option.)
Ultimate RECALL Commands

(Options) the following options are available:

H  suppresses the tape label. (This is equivalent to the HDR-SUPP or SUPP modifiers.)

I  suppresses item-ID listing. (This is equivalent to the ID-SUPP modifier.)

P  routes output to spooler. (This is equivalent to the LPTR modifier.)

Description

The S-DUMP command dumps selected file items to tape in a sorted sequence. The T-DUMP command performs the same function without sorting the selected items.

A tape label containing the filename, tape block (record) length in hexadecimal, and the current time and date is written at the start of the dump, before any items. You can suppress the heading by specifying the HDR-SUPP modifier or the H option. You can specify additional information for the header by using the HEADING modifier.

If dictionary items are being dumped, file definition items are not dumped.

After all sorted items have been dumped, an EOF mark is written to tape. The last record may be filled with pad characters after the end of valid data. The pad character for S-DUMP is the SB character (X'FB'), which prints as \.

The tape should be explicitly attached by the T-ATT command before the S-DUMP command is issued.
S-DUMP

```
: T-ATT
attaches the tape using default settings.

Tape 0 attached
Block size: 8192

:S-DUMP CUSTOMERS HEADING "S-DUMP Format"

Block size: 8192
1 1
2 2
3 3
4 4
5 5
5 items dumped.

:T-REW
rewinds the tape to display the tape records.
Block size: 8192
Rewinding...

:T-READ
displays the tape records.
Block size: 8192


Record = 1

  1 "Quality Lighting Products" Mat" 5454 W. Mariposa D
  51 owney"CA" 91710"3015551234"  2 "Service Office Prod
  10 ts"Lee" 4512 Orange"Santa Ana" CA" 92222"7145551234"
  151 3 "Universal Copiers" Marina" 211 Westgate" Long Beach
  201 "CA" 91832"2135551234"  4 "Kelley Brothers" Jerry" 1234
  251 5 Main Street"Anaheim" CA" 92006"7145556789"  5 "Serv
  301 ce Office Products" Shelby" 3114 Paradise Drive" San
  351 Fernando"CA" 91340"8185551234"  3 X: [X] [X] [X] [X] [X]
  401 [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X]
  451 [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X]
  501 [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X]

...

8051 [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X]
810: [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X]
8151 [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X] [X]

End of file:
```
SELECT

The SELECT command creates a select-list.

Syntax

```
SELECT filename {itemlist} {sel-criteria} {output-specifications} {(W)
```

- **filename**  
  name of file from which items are to be selected.

- **itemlist**  
  specifies items to select; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.

- **sel-criteria**  
  provides the tests that the items chosen for consideration must pass in order to be included in the selection. If no selection criteria are present, all items being considered are selected.

- **output-specifications**  
  indicates the attributes to be placed in the select-list; if not specified, the item-IDs are placed in the select-list.

*Note:*  
SELECT does not use default output specifications.

(W  
allows BASIC subroutine to write to files it opens.

Description

The SELECT command selects items from the specified file and stores the information in a temporary list until the next command is finished executing. After the select-list has been processed (or at the end of the next command's execution), the temporary select-list is released and is no longer available.

To save the list for future processing, execute the SAVE-LIST command as the next command after the SELECT command.

If the next command does not use the select list, or if the command contains an error, the select-list is lost and must be selected again.
If output specifications are used, the select-list will contain attribute data. If no output specifications are used, the select-list will contain item-IDs. The items are not resequenced; they are stored in the select-list in the same order as they are stored in the file.

```
:SELECT INVOICE WITH FLAGALL = "0"

4 items selected.

:SORT INVOICE COMPANY TOT.PRICE

PAGE 1

INVOICE... Company Name....... Tot Price.
1681 Kelley Brothers $269.50
1682 Kelley Brothers $84.35
1686 Quality Lighting Prod $220.95
1696 Universal Copiers $55.80

4 items listed.
```
SORT

The SORT command rearranges selected items into a specified sequence, then generates formatted output of selected items and attributes.

Syntax

SORT filename {itemlist} {sel-criteria} {sort-criteria} {output-specifications {print-limiters}} {modifiers} {(options}

filename name of file to be sorted and listed.

itemlist specifies items to sort; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.

sel-criteria provides the tests that the items chosen for consideration must pass in order to be included in the output. If no selection criteria are present, all items being considered are selected.

sort-criteria specifies attributes to use as sort keys to resequence the items for output. A number of sort criteria can be used in order to create a multi-level sorted report. The sort can be in ascending or descending order. If no sort criteria are present, the items are sorted by item-ID in ascending order.

output-specifications indicates which attributes are to be listed; these determine which fields of information are included on the report, and how they are formatted.

print-limiters indicates criteria that a value in the specified attribute must meet in order to be included on the report. If no print-limiters are present, all the values will be included for a multivalued attribute.

modifiers control the format and layout of the report. Any modifier described in Chapter 2, except WITHIN, can be included.

(options the following options are available:
A  alignment; forms only.
B  background; forms only.
C  if the report is columnar, suppresses the page heading line and end message and also omits the column heading line that identifies the attribute names in the report. If the report is non-columnar, suppresses only the page heading and end message; the column headings are not suppressed. (This is equivalent to the COL-HDR-SUPP modifier.)
D  suppresses all detail lines on report. Any break lines and totals specified in the Ultimate RECALL command are output. (This is equivalent to the DET-SUPP modifier.)
H  suppresses the page heading line (time and date on the left, page number on the right) and the "n items listed" message at the end of the report. (This is equivalent to the HDR-SUPP or SUPP modifiers.)
I  suppresses item-ID listing. (This is equivalent to the ID-SUPP modifier.)
M  specifies multiple items per page; forms only.
N  suppresses wait at end of page. (This is equivalent to the NOPAGE modifier.)
P  routes output to the spooler. (This is equivalent to the LPTR modifier.)
W  allows BASIC subroutine to write to files it opens.
Z  resets page number to 1 for each form; forms only.

Description

The SORT command displays or prints selected data from selected items in a sorted order. The LIST command performs the same report function but lists the items in the order they are stored in the file.

If no sort-criteria are specified, the sort is in ascending order and the item-IDs are used as sort keys. If multiple sort keys are present, the primary sort is on the first specified key. The sort proceeds in the order in which the sort criteria are listed. The final sort key is the item-ID, which is always used in a sort, even when other sort criteria are specified. In generating the values used in the sort key comparison.
correlatives in the attribute definition are processed, but conversion specifications are not.

The command sorts attributes based on the V/TYP code in the attribute definition item. For more information, see the discussion of V/TYP in Chapter 4.

After the selected file items are sorted, the specified attributes in that item are output according to the output specifications and modifiers.

If forms expressions are used, output is formatted according to the forms expressions. If no forms expressions are present, a columnar format is used if the number of attributes and their names being listed fit in the output page width, otherwise a non-columnar format is used. (Columnar and non-columnar formats are described in Chapter 1. Forms expressions are described in Chapter 6.)
**SORT**

:SORT INVOICE WITH FLAGALL = "O" BY DATE COMPANY.L DATE TOTAL TOT.PRICE

<table>
<thead>
<tr>
<th>COMPANY NAME</th>
<th>DATE</th>
<th>TOTAL PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal Copiers</td>
<td>07/01/92</td>
<td>$55.80</td>
</tr>
<tr>
<td>Quality Lighting Products</td>
<td>08/21/92</td>
<td>$220.95</td>
</tr>
<tr>
<td>Kelley Brothers</td>
<td>08/23/92</td>
<td>$269.50</td>
</tr>
<tr>
<td>Kelley Brothers</td>
<td>08/24/92</td>
<td>$84.35</td>
</tr>
</tbody>
</table>

4 items listed.

:SORT INVOICE WITH FLAGALL = "O" BY-EXP PROD.NO > "7000" PROD.NO BREAK-ON DESC.L "L" TOTAL EXT.PRICE DET-SUPP

<table>
<thead>
<tr>
<th>PROD.NO</th>
<th>DESCRIPTION</th>
<th>EXT PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>7001</td>
<td>Water</td>
<td>$11.10</td>
</tr>
<tr>
<td>7015</td>
<td>Raspberry Soda</td>
<td>$10.40</td>
</tr>
<tr>
<td>7017</td>
<td>Wild Cherry Soda</td>
<td>$10.40</td>
</tr>
<tr>
<td>7055</td>
<td>Diet Cola</td>
<td>$57.60</td>
</tr>
<tr>
<td>7056</td>
<td>Regular Cola</td>
<td>$29.85</td>
</tr>
<tr>
<td>7065</td>
<td>Diet Root Beer</td>
<td>$19.90</td>
</tr>
<tr>
<td>7066</td>
<td>Root Beer</td>
<td>$19.80</td>
</tr>
<tr>
<td>7331</td>
<td>Tomato Juice</td>
<td>$19.20</td>
</tr>
<tr>
<td>7345</td>
<td>Apple Juice</td>
<td>$77.00</td>
</tr>
<tr>
<td>8036</td>
<td>Cookies</td>
<td>$40.20</td>
</tr>
<tr>
<td>8123</td>
<td>Crackers</td>
<td>$20.60</td>
</tr>
</tbody>
</table>

19 items listed.
SORT-ITEM

The SORT-ITEM command rearranges selected items into a specified sequence, then lists their attributes.

Syntax

SORT-ITEM  filename  {itemlist}  {sel-criteria}  {sort-criteria}  {modifiers}  {options}

filename  name of file to be listed.
itemlist  specifies items to list; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.

sel-criteria  provides the tests that the items chosen for consideration must pass in order to be included in the list. If no selection criteria are present, all items being considered are selected.

sort-criteria  specifies attributes to use as sort keys to resequence the items for output. A number of sort criteria can be used in order to create a multi-level sorted report. The sort can be in ascending or descending order. If no sort criteria are present, the items are sorted by item-ID in ascending order.

modifiers  control the format and layout of the output. The following modifiers described in Chapter 2 are available:

DBL-SPC  doublespaces output.
FOOTING  displays specified footing.
HDR-SUPP  suppresses the page heading line. (This is equivalent to the H option.)
HEADING  displays specified heading.
ID-SUPP  suppresses item-ID listing. (This is equivalent to the I option.)
LPTR  routes output to spooler. (This is equivalent to the P option.)
**Description**

The SORT-ITEM command is similar to the CT command, but it provides many of the Ultimate RECALL capabilities, such as selection criteria and headings or other output formatting. It also provides a sorting capability.

The entire contents of the selected items are displayed in the specified sorted order on the terminal or printer. Attribute numbers are displayed at the left margin.

No correlative or conversion code processing takes place.

---

**SORT-ITEM**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOPAGE</td>
<td>no end-of-page waiting. (This is equivalent to the N option.)</td>
</tr>
<tr>
<td>SUPP</td>
<td>suppresses the page heading line. (This is equivalent to the H option.)</td>
</tr>
<tr>
<td>TAPE</td>
<td>obtains items from tape.</td>
</tr>
</tbody>
</table>

The following options are available:

- F forces a new page after every item.
- H suppresses the page heading line. (This is equivalent to the HDR-SUPP or SUPP modifiers.)
- I suppresses item-ID listing. (This is equivalent to the ID-SUPP modifier.)
- N suppresses wait at end of page. (This is equivalent to the NOPAGE modifier.)
- P routes output to the spooler. (This is equivalent to the LPTR modifier.)
- S suppresses display of line numbers.
- X displays output in hexadecimal.
:SORT-ITEM INVOICE WITH ZIP = "917"]"
SORT-LABEL

The SORT-LABEL command generates sorted and formatted output of data. Item data can be grouped into blocks, with several blocks placed across the page as in a set of mailing labels.

Syntax

SORT-LABEL filename {itemlist} {sel-criteria} {sort-criteria} {output-specifications} {print-limiters} {modifiers} {options}

filename name of file to be listed.

itemlist specifies items to list; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.

sel-criteria provides the tests that the items chosen for consideration must pass in order to be included in the list. If no selection criteria are present, all items being considered are selected.

sort-criteria specifies attributes to use as sort keys to resequence the items for output. A number of sort criteria can be used in order to create a multi-level sorted report. The sort can be in ascending or descending order. If no sort criteria are present, the items are sorted by item-ID in ascending order.

output-specifications indicates which attributes are to be listed; these determine which fields of information are included on the report, and how they are formatted.

print-limiters indicates criteria that a value in the specified attribute must meet in order to be included on the report. If no print-limiters are present, all the values will be included for a multivalued attribute.

modifiers control the format and layout of the report. Any modifier described in Chapter 2, except WITHIN, can be included.
(options

the following options are available:

C suppresses the page heading line and end message;
also suppresses top-of-forms.

D suppresses all detail lines on report. Any break lines
and totals specified in the Ultimate RECALL
command are output. (This is equivalent to the DET-
SUPP modifier.)

H suppresses the page heading line (time and date on
the left, page number on the right) and the "n items
listed" message at the end of the report. (This is
equivalent to the HDR-SUPP or SUPP modifier.)

I suppresses item-ID listing. (This is equivalent to the
ID-SUPP modifier.)

N suppresses wait at end of page. SORT-LABEL output
is generated as one continuous report page; all
headers but the first is suppressed. (This is
equivalent to the NOPAGE modifier.)

P routes output to the spooler. (This is equivalent to
the LPTR modifier.)

W allows BASIC subroutine to write to files it opens.

Description

Before searching the file and creating the list, the SORT-LABEL command
displays a question mark (?) to prompt for an additional set of
parameters. You must enter the information needed to create the desired
arrangement of attributes and blocks per line in the following order:

count, rows, skip, indent, size, space{(C}{S})

where

count number of items (labels) across the page.
rows number of lines printed per label (height of each label, in
rows).
skip number of lines to skip between each label (vertical spacing between labels, in rows).
indent number of spaces to indent from the left margin (to allow
for printing specified text).
size maximum number of characters to be printed in an
attribute (label width, in columns).
space  number of spaces between items (horizontal spacing between labels, in columns).

C  specifies that null attributes are not to be printed (if omitted, null values are printed as all blanks).

S  specifies that a new line of labels is to start after each control break (used with the BREAK-ON connective).

The values used must conform to the range:

\[(\text{count} \times (\text{size} + \text{space}) + \text{indent}) \leq \text{page width}\]

where page width is the number defined in the TERM command for the current output device (printer or terminal). If the maximum number of characters specified exceeds the page width, the system displays a message similar to the following, where \(n\) is the invalid parameter:

\[\text{The range of the parameter "n" is not acceptable}\]

If indent is non-zero, the command displays a question mark (?) to prompt for the text to be printed. One question mark is displayed for each row specified for the parameter rows. When the listing is printed, this text is displayed in the indent area. (To specify null text for a line, press RETURN at the corresponding prompt.)

If indent is zero or null, no text is displayed at the left margin; instead the first block of data begins at column 1.

The standard heading is displayed at the top of each page, unless suppressed by the COL-HDR-SUPP or HDR-SUPP modifier or the C or H option.

If headings are suppressed by the COL-HDR-SUPP modifier or the C option, pagination and all top-of-forms are also suppressed. This produces a continuous forms format without page breaks.

The SORT-LABEL command ignores V/TYP and V/MAX. All data is printed left justified using as many characters as specified in the size parameter.
**Ultimate RECALL Commands**

```
:SORT-LABEL CUSTOMERS BY ZIP (CI 2, 4, 2, 13, 24, 6
COMPANY CONTACT STREET CITY, STATE
Service Office Products Shelby 3114 Paradise Drive San Fernando, CA 91340
Quality Lighting Products Mat 5454 W. Mariposa Downey, CA 91710
Universal Copiers Marina 211 Westgate Long Beach, CA 91832
Kelley Brothers Jerry 12345 Main Street Anaheim, CA 92006
Service Office Products Lee 4512 Orange Santa Ana, CA 92222
```
The **SREFORMAT** command creates new items by reformatting existing items. The new items can be placed in a separate file, in the current file, or on tape. If the items are output to tape, they are in sorted order.

**Syntax**

```plaintext
SREFORMAT filename {itemlist} {sel-criteria} {sort-criteria}
{output-specifications} {print-limiters} {modifiers}
{(options)}
```

- **filename**
  - name of file that contains items to be reformatted.

- **itemlist**
  - specifies items to reformat; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.

- **sel-criteria**
  - provides the tests that the items chosen for consideration must pass in order to be reformatted. If no selection criteria are present, all items being considered are selected.

- **sort-criteria**
  - specifies attributes to use as sort keys to resequence the items for output. A number of sort criteria can be used in order to create a multi-level sorted report. The sort can be in ascending or descending order. If no sort criteria are present, the items are sorted by item-ID in ascending order.

- **output-specifications**
  - indicates the attributes that are to be used in the new item; if not specified and there are no default output specifications, no items are created.

- **print-limiters**
  - indicates criteria that a value in the specified attribute must meet in order to be reformatted. If no print-limiters are present, all the values are included for a multivalued attribute.
modifiers control the format and layout of the output. The following modifiers described in Chapter 2 are available:

**COL-HDR-SUPP** suppresses the tape label; meaningful only if destination is tape. (This is equivalent to the C option.)

**HDR-SUPP** suppresses the tape label; meaningful only if destination is tape. (This is equivalent to the H option.)

**HEADING "text"** specifies text to be included in the tape label; meaningful only if destination is tape.

**ID-SUPP** suppresses item-ID; causes the reformatting to skip the first specified attribute and use the second attribute as the item-ID of the new item. (This is equivalent to the I option.)

**LPTR** when the destination is to tape, routes the list of item-IDs to the spooler; meaningful only if destination is tape. (This is equivalent to the P option.)

**TAPE** gets data from file on tape.

(options) the following options are available:

**C** suppresses the tape label; meaningful only if destination is tape. (This is equivalent to the COL-HDR-SUPP modifier.)

**H** suppresses the tape label; meaningful only if destination is tape. (This is equivalent to the HDR-SUPP or SUPP modifiers.)

**I** suppresses item-ID; causes the reformatting to skip the first specified attribute and use the second attribute as the item-ID of the new item. (This is equivalent to the ID-SUPP modifier.)

**P** when the destination is to tape, routes the list of item-IDs to the spooler; meaningful only if destination is tape. (This is equivalent to the LPTR modifier.)

**W** allows BASIC subroutine to write to files it opens.
Description

When the SREFORMAT command is invoked, it prompts for the destination filename:

File name>

- to store the reformatted file in a different file on disk, enter the filename.
- to write the file to tape, enter the word TAPE.
- to store the reformatted items back into the current file, press RETURN.

Note: When reformatting to the current file, you should specify an itemlist or use a select-list; otherwise, the command may execute an infinite loop.

The value of the first attribute defined by the output specifications is used as the item-ID. The remaining attributes make up the item. The items are output in sorted order. However, the sorted order is preserved only when outputting to tape. When you write the items to disk, they are placed on the disk according to the hashing value of their item-IDs.

Reformatting to Tape

When the reformatting is to tape, a tape label containing the filename, tape block (record) length in hexadecimal, and the current time and date is written at the start of the dump, before any items. You can suppress the heading by specifying the HDR-SUPP modifier or the H option. You can specify additional information for the header by using the HEADING modifier.

Before executing the command, you should issue a T-ATT. Specify the blocksize equal to the maximum length of the reformatted items. As each item is reformatted, the item-ID and attributes for the new item are concatenated, and either truncated or padded at the end with nulls (X'00') to obtain a block the same length as the blocksize specified by the T-ATT command.

One tape block is written for each item. Item-IDs from the file being reformatted are displayed as the items are written to tape unless the ID-SUPP (I option) is specified. After all the data has been written, two EOFs are written to terminate the tape.
Ultimate RECALL Commands

This option is intended primarily to create tapes to be used with non-Pick systems. Each attribute should be formatted to a fixed length and the tape should be attached at a blocksize equal to the sum of all the lengths of the attributes plus the length of the item-ID.

:T-ATT 0,100
Tape 0 attached
Block size: 100

:SREFORMAT CUSTOMERS BY COMPANY COMPANY NAME ADDRESS CITY STATE ZIP (H
File name>TAPE
Block size: 100
1 4
2 1
3 2
4 5
5 3
5 items dumped.

:T-REW
Block size: 100
Rewinding...

:T-READ
Block size: 100

Record = 1
1 Kelley Brothers Jerry
51 12345 Main Street Ananeim CA 92006

Record = 2
1 Quality Lighting Products Mat
51 5454 W. Mariposa Downey CA 91710

End of file

creates a file on tape with company as the first value, followed by name, address, city, state, and zip. The length of each tape block is 100 characters.
The SSELECT command creates a select-list sorted into a specified order.

**Syntax**

```
SSELECT filename {itemlist} {sel-criteria} {sort-criteria} {output-specifications} {((W)
```

- **filename**
  - name of file from which items are to be selected.

- **itemlist**
  - specifies items to select; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.

- **sel-criteria**
  - provides the tests that the items chosen for consideration must pass in order to be included in the selection. If no selection criteria are present, all items being considered are selected.

- **sort-criteria**
  - specifies attributes to use as sort keys to resequence the items for output. A number of sort criteria can be used in order to create a multi-level sorted report. The sort can be in ascending or descending order. If no sort criteria are present, the items are sorted by item-ID in ascending order.

- **output-specifications**
  - indicates the attributes to be placed in the select-list; if not specified, the item-IDs are placed in the select-list.

**Note:** *SSELECT does not use default output specifications.*

- **(W)***
  - allows BASIC subroutine to write to files it opens.

**Description**

The SSELECT command selects items from the specified file, sorts them into the specified order, then stores the information in a temporary list until the next command is finished executing. After the select-list has been processed (or at the end of the next command's execution), the temporary select-list is released and is no longer available.
To save the list for future processing, execute the SAVE-LIST command as the next command after the SSELECT command.

If the next command does not use the select list, or if the command contains an error, the select-list is lost and must be selected again.

If output specifications are used, the select-list will contain attribute data. If no output specifications are used, the select-list will contain item-IDs. If the BY-EXP modifier is used, each element in the select-list (item-ID or attribute data) is followed by a value mark, then the number of the value within the attribute (VMC). Such lists should only be used by list-type commands, not sort commands.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SSELECT INVOICE BY-EXP PROD.NO</td>
<td>Select items in order by product number.</td>
</tr>
<tr>
<td>52 items selected.</td>
<td></td>
</tr>
<tr>
<td>:SAVE-LIST PROD.NO</td>
<td>Saves list for future use.</td>
</tr>
<tr>
<td>'PROD.NO' saved - 1 frames used.</td>
<td></td>
</tr>
<tr>
<td>:GET-LIST PROD.NO</td>
<td>Gets previously saved list.</td>
</tr>
<tr>
<td>52 items selected.</td>
<td></td>
</tr>
<tr>
<td>:LIST INVOICE PROD.NO DESC</td>
<td>The list is in sorted order by product number.</td>
</tr>
<tr>
<td>PAGE 1</td>
<td>11:03:46 30 JUN 1992</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1681 C005</td>
<td>Herb Tea</td>
</tr>
<tr>
<td>1683 C005</td>
<td>Herb Tea</td>
</tr>
<tr>
<td>1686 C005</td>
<td>Herb Tea</td>
</tr>
<tr>
<td>1681 2025</td>
<td>Regular Tea</td>
</tr>
<tr>
<td>1683 2025</td>
<td>Regular Tea</td>
</tr>
<tr>
<td>1686 2025</td>
<td>Regular Tea</td>
</tr>
<tr>
<td>1687 2025</td>
<td>Regular Tea</td>
</tr>
</tbody>
</table>

52 items listed.
The STAT command provides a count of the number of items selected. If an attribute is specified, the command accumulates and reports the total value of that attribute and its average value. If no attribute is specified, the command accumulates and reports the total number of bytes in the selected items and the average number per item.

**Syntax**

```
STAT filename {itemlist} {sel-criteria} {attrib-name} {((P})
```

- **filename** specifies file to be totaled.
- **itemlist** specifies items to totaled; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.
- **sel-criteria** conditions that must be met in order for the item to be included in the calculations. If no selection criteria are present, all items being considered are selected.
- **attrib-name** name of attribute to be totaled. If the attribute number of attrib-name is 9999, or if no attrib-name is specified, the total number of bytes in the entire item is accumulated.
- **(P** routes output to spooler. (This is equivalent to the LPTR modifier.)

**Description**

The STAT command selects the items to be used, based on the selection criteria. Each item is totaled as specified in the command and the total is accumulated. A count is also kept of each item selected and used.

Correlatives are processed before accumulating the values; any conversions present are applied to the total and average just before they are printed.

If an attribute is specified, the attribute definition item should reference an actual attribute. If a dummy attribute (other than 9999) is referenced, zero is returned.
Attribute marks are included in the statistics. When an entire item is processed, the count field, item-ID, and first attribute mark are included in the calculations.

When all selected items have been processed, an average value is calculated by dividing the total by the count of items. The command outputs the statistics in a format similar to the following:

Statistics of name:
Total = t  Average = a  Count = c

where
name  attribute name, if specified; otherwise filename.
t  total number of bytes in the attribute or items.
a  average number of bytes.
c  number of items.

The STAT command can be used as an alternative to the SUM command when the number of items selected and the attribute's average value per item is needed in addition to the total value.

:STAT INVOICE  Accumulates totals for the entire file.
Statistics of INVOICE:
Total = 1226  Average = 204.33  Count = 6

:STAT INVOICE PRICE.L  Accumulates totals for the attribute PRICE.
Statistics of Price:
Total = 5636.33  Average = 939.3883  Count = 6
SUM

The SUM command accumulates and reports the total of all the values of one attribute name for a selected set of file items.

Syntax

SUM filename {itemlist} {sel-criteria} {attrib-name} {(P}

filename specifies file to be totaled.

itemlist specifies items to be totaled; item-IDs should be enclosed in single quotes (''). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.

sel-criteria conditions that must be met in order for the item to be included in the calculations. If no selection criteria are present, all items being considered are selected.

attrib-name name of attribute to be totaled; if the attribute number of attrib-name is 9999, or if no attrib-name is specified, the total number of bytes in the entire item is accumulated.

(P routes output to spooler. (This is equivalent to the LPTR modifier.)

Description

The SUM command selects the items to be used, based on the selection criteria. Each item is totaled as specified in the command and the total is accumulated.

Correlatives are processed before accumulating the values; any conversions present are applied to the total and average just before they are printed.

If an attribute is specified, the attribute definition item should reference an actual attribute. If a dummy attribute (other than 9999) is referenced, zero is returned.

Attribute marks are included in the statistics. When an entire item is processed, the count field, item-ID, and first attribute mark are included in the calculations.
When all selected items have been processed, the command outputs the statistics in the format:

\[
\text{Total of name} = t
\]

where

name \hspace{1cm} \text{attribute name, if specified; otherwise filename.}

\( t \hspace{1cm} \text{total value of attribute or items.} \)

The SUM command can be used as an alternative to the STAT command when you need the total value but not the number of items selected and the attribute's average value per item.

```
:SUM INVOICE
Accumulates totals for the entire file.
Total of INVOICE = 1226

:SUM INVOICE PRICE.L
Accumulates totals for the attribute PRICE.L.
Total of Price = 9636.35
```
The T-DUMP command dumps selected file items to the tape.

Syntax

T-DUMP filename {itemlist} {sel-criteria} {modifiers} {{(options)}}

filename: name of file to be dumped.

itemlist: specifies items to dump; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.

sel-criteria: provides the tests that the items chosen for consideration must pass in order to be included in the dump. If no selection criteria are present, all items being considered are selected.

modifiers: control the format and layout of the output. The following modifiers described in Chapter 2 are available:

- HDR-SUPP: suppresses the tape label. (This is equivalent to the H option.)
- HEADING "text": specifies text to be included in the tape label.
- ID-SUPP: suppresses item-ID listing. (This is equivalent to the I option.)
- LPTR: routes the list of item-IDs to the spooler. (This is equivalent to the P option.)

(options): the following options are available:

- H: suppresses the tape label (This is equivalent to the HDR-SUPP or SUPP modifiers.)
- I: suppresses item-ID listing. (This is equivalent to the ID-SUPP modifier.)
- P: routes the list of item-IDs to the spooler. (This is equivalent to the LPTR modifier.)
Ultimate RECALL Commands

Description

The T-DUMP command dumps selected items to tape. The S-DUMP command performs the same function, but first sorts the selected items.

A tape label containing the filename, tape block (record) length in hexadecimal, and the current time and date is written at the start of the dump, before any items. You can suppress the heading by specifying the HDR-SUPP modifier or the H option. You can specify additional information for the header by using the HEADING modifier.

If dictionary items are being dumped, file definition items are not dumped.

After all items have been dumped, an EOF mark is written to tape. The last record may be filled with pad characters after the end of valid data. The pad character for T-DUMP is the SB character (X'FB'), which prints as [.

The tape should be explicitly attached by the T-ATT command before the T-DUMP command is issued.
• Tape 0 attached
  Block size: 8192

• T-DUMP INVOICE
  Block size: 8192
  1 1686
  2 1683
  3 1687
  4 1696
  5 1681
  6 1682
  6 items dumped.

• T-REW
  Block size: 8192
  Rewinding...

• T-READ
  Block size: 8192


Record = 1
  1 1686~Quality Lighting Products~9000~7056~7055~7066
  51 7065~2025~6032~0005~3007~3035~3004~5003~8123
  101 "212~1|1|4|1|2|1|1|1|1|1|8989|8999|9989|9989|9989|9989|9989|9989|9989\O\O\O\O\O\O\O\O~Mat~5454\W\Mariposa~917~\O\Net
  251 10~Truck~301551234~\1683~Service Office Products
  301 ~8988~7345~7331~8123~6032~6068~0005~2025~8036~4241
  351 :5003~5005~3~3~3~3~2|4|\~\1~8892~8892~8892~

  8051 :\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\n
End of file
T-LOAD

The T-LOAD command loads specified file items from the tape attached to the terminal.

Syntax

```
T-LOAD filename {itemlist} {sel-criteria} {modifiers}
{options}
```

**filename**  
name of file into which items are to be loaded; file must already exist.

**itemlist**  
specifies items to load; item-IDs should be enclosed in single quotes ('). If no itemlist is present, all items are considered, unless a select-list has been created by the previous command. If a select-list is present, and no itemlist is present, only the item-IDs in the select-list are considered.

**sel-criteria**  
provides the tests that the items chosen for consideration must pass in order to be loaded. If no selection criteria are present, all items being considered are selected.

**modifiers**  
control the format and layout of the output. The following modifiers described in Chapter 2 are available:

- **ID-SUPP**  
suppresses item-ID listing. (This is equivalent to the I option.)

- **LPTR**  
routes the list of item-IDs to the spooler. (This is equivalent to the P option.)

**(options)**  
the following options are available:

- **I**  
suppresses item-ID listing. (This is equivalent to the ID-SUPP modifier.)

- **O**  
overwrites item in file if item on tape has same item-ID.

- **P**  
routes the list of item-IDs to the spooler. (This is equivalent to the LPTR modifier.)

Description

The T-LOAD command loads selected items saved by a T-DUMP or S-DUMP command.
The tape should be explicitly attached by the T-ATT command before the T-LOAD command is issued.

The tape should be positioned at the first block (record) of the file to be loaded. The system tape handler does not need to read the tape label in order to determine block size; it automatically determines the block size from the tape itself. If the tape is not positioned at the first block, the first item may be incorrect and may be processed improperly because the tape blocks and items are not aligned.

The T-LOAD command selects the items from the tape according to the specified criteria, if any.

If the O option is present, items on the tape whose item-IDs match item-IDs in the destination file overwrite the items in the destination file. Item-IDs are listed on the terminal as they are loaded unless the I option is specified.

```
: T-LOAD INVOICE
Block size: 8192

1 1683
2 1686
3 1687
4 1696
5 1681
6 1682
6 item(s) loaded.
```
4 Attribute Definition Items

Attribute definition items determine how data is displayed and sorted and are defined in the dictionary portion of the file. The attribute definition items are used by Ultimate RECALL after it has accessed the file and item specified in the statement.

The following can be defined in an attribute definition item:

- the attribute’s location (attribute number) in the data file.
- the column heading to use on reports.
- the structure code if the attribute controls or is controlled by other attributes.
- the output conversion format, such as date, money, or time formats.
- the intermediate processing code (correlative).
- left or right justification in the report column; the justification also determines alphanumeric or numeric sorting for the attribute.
- maximum number of characters to be displayed on a line.

Table 4-1 shows the format of attribute definition items. The subsections following the table give a detailed description of each attribute.
### Table 4-1. Attribute Definition Item Format

<table>
<thead>
<tr>
<th>Attribute Number</th>
<th>Standard Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>item-ID</td>
<td>attribute name.</td>
</tr>
<tr>
<td>001</td>
<td>D/CODE</td>
<td>definition code; can be one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A attribute definition item.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S synonym definition item.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>U Ultimate UPDATE definition item.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X skip attribute output.</td>
</tr>
<tr>
<td>002</td>
<td>A/AMC</td>
<td>attribute number, sequential position in data item.</td>
</tr>
<tr>
<td>003</td>
<td>S/NAME</td>
<td>column heading.</td>
</tr>
<tr>
<td>004</td>
<td>S/AMC</td>
<td>structure code.</td>
</tr>
<tr>
<td>005</td>
<td></td>
<td>reserved.</td>
</tr>
<tr>
<td>006</td>
<td></td>
<td>reserved.</td>
</tr>
<tr>
<td>007</td>
<td>V/CONV</td>
<td>conversion code.</td>
</tr>
<tr>
<td>008</td>
<td>V/CORR</td>
<td>correlative code.</td>
</tr>
<tr>
<td>009</td>
<td>V/TYP</td>
<td>justification; can be one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>L left.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R right.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RN right, numeric only.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T text.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>U text, no wrapping.</td>
</tr>
<tr>
<td>010</td>
<td>V/MAX</td>
<td>maximum number of characters to display on a line.</td>
</tr>
</tbody>
</table>
Components of Attribute Definition Items

This section contains a detailed description of each attribute of an attribute definition item. The item-ID and attributes 1, 2, 9, and 10 are required to guarantee that the definition items work properly.

Item-ID

The item-ID of an attribute definition item defines the attribute name in the data. Ultimate RECALL uses this attribute name to identify the attribute.

The item-ID of an attribute definition item in a file's dictionary can be a number or a name of up to 50 characters.

Consecutively numbered item-IDs, starting with 1, can be defined and are used to format and output Ultimate RECALL statements that do not specify attribute names to include on the report. These attribute definition items are used as default output specifications. For more information, see the section, Output Specifications, in Chapter 2.

Some characters have special meaning to the system and should not be used in item-IDs for either dictionary or data items. These characters are listed in Table 4-2. Any other character can be used in an item-ID.

The following are all valid item-IDs:

1
21
ADDR
PRICE
QUANTITY-ON-HAND
# Attribute Definition Items

## Table 4-2. Characters Not Recommended for Item-IDs

<table>
<thead>
<tr>
<th>Name</th>
<th>Decimal Value</th>
<th>Hexadecimal Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUL</td>
<td>0</td>
<td>00</td>
<td>system delimiter for ascending sort keys.</td>
</tr>
<tr>
<td>SOH</td>
<td>1</td>
<td>01</td>
<td>system delimiter for item-IDs in indexes.</td>
</tr>
<tr>
<td>[</td>
<td>91</td>
<td>5B</td>
<td>wild card.</td>
</tr>
<tr>
<td>]</td>
<td>93</td>
<td>5D</td>
<td>wild card.</td>
</tr>
<tr>
<td>^</td>
<td>94</td>
<td>5E</td>
<td>wild card.</td>
</tr>
<tr>
<td>~</td>
<td>126</td>
<td>7E</td>
<td>system delimiter for descending sort keys.</td>
</tr>
<tr>
<td>SB</td>
<td>251</td>
<td>FB</td>
<td>system delimiter for sort fields, also used in T-DUMP tape format.</td>
</tr>
<tr>
<td>SVM</td>
<td>252</td>
<td>FC</td>
<td>system delimiter used as sub-value mark.</td>
</tr>
<tr>
<td>VM</td>
<td>253</td>
<td>FD</td>
<td>system delimiter used as value mark.</td>
</tr>
<tr>
<td>AM</td>
<td>254</td>
<td>FE</td>
<td>system delimiter used as attribute mark.</td>
</tr>
</tbody>
</table>
Components of Attribute Definition Items

The D/CODE (attribute 1) specifies the type of item being defined. The following codes can be used for an attribute definition item:

- **A** attribute definition item.
- **S** synonym attribute definition item. Ultimate RECALL treats it as an A code. Retained for compatibility with other releases.
- **U** Ultimate UPDATE definition item. These items are explained in the Ultimate UPDATE Reference Guide.
- **X** skip attribute output. An X code specifies no output of this attribute. Typically used with numbered attribute definition items that are defined as default output specifications to skip an attribute, but continue the search for the next consecutively numbered item-ID. An attribute definition item with an X as its D/CODE cannot be specified in an Ultimate RECALL statement.

The A/AMC attribute (attribute 2) is the sequential position of this attribute in the data items; for example, a 1 indicates the first attribute. The A/AMC acts as a pointer for Ultimate RECALL to access the attribute's value in each data item.

An AMC of 0 is used to reference item-IDs in the data file; this allows item-IDs to be accessed via different names and to be displayed under different column headings, if desired.

A dummy AMC (that is, an AMC that is greater than the highest actual AMC on the file) can be used if the attribute has a correlative (calculated, not stored) value.

An AMC of 9998 is used to access the current item counter (item sequence number in the report). An AMC of 9999 is used to access the SIZE or byte (character) count field of the item.

The S/NAME attribute (attribute 3) is an optional field. If used, it contains the column heading to use for this attribute or Ultimate RECALL reports and listings. If no S/NAME is specified, the attribute definition item-ID is used as the column heading.

For example, if an attribute definition item called DESC had no S/NAME and you listed the attribute, the heading would be DESC. If the attribute
Attribute Definition Items

definition item contained a S/NAME of Description and you listed the file, the heading would be Description.

To suppress printing of a column heading, specify a backslash character (\) as the S/NAME.

To specify a multiple-line heading, separate each line with a value mark. For example, to have Invoice Number display on two lines, enter it as follows:

    Invoice\Number

It will be displayed as

    Invoice...
    Number

S/AMC - Structure Code

The S/AMC attribute (attribute 4) is an optional field. It is null unless the attribute has a controlling or dependent relationship with other attributes. A controlling and dependent relationship means that a set of attributes are listed together or associated for other Ultimate RECALL purposes. Typical uses are for invoices, sales orders, inventories, or checks.

In a report, the controlling attribute is always listed first and the dependent attributes are listed in the order they are specified in the S/AMC attribute for the controlling attribute, not in the order they are specified in the Ultimate RECALL statement. An asterisk is displayed under the column heading of each dependent attribute. If the attributes are specified as part of the output specifications, the dependent attributes are listed only if the controlling attribute is specified.

Attribute 4 contains the structure code that associates the (one only) controlling attribute and its dependent attributes. A controlling attribute has the following structure code format:

    C: amc1; amc2 ...

where amcn is the attribute number or name of the nth dependent attribute.

A dependent attribute has the following structure code format:

    D: amc
Components of Attribute Definition Items

where amc is the controlling attribute number or name.

<table>
<thead>
<tr>
<th>PROD.NO</th>
<th>DESC</th>
</tr>
</thead>
<tbody>
<tr>
<td>001 A</td>
<td>001 A</td>
</tr>
<tr>
<td>002 3</td>
<td>002 991</td>
</tr>
<tr>
<td>003 Product</td>
<td>Number 003 Description</td>
</tr>
<tr>
<td>004 C;DESC;PRICE;QTY 004 D;PROD.NO</td>
<td></td>
</tr>
<tr>
<td>DEL.DATE</td>
<td>_ _</td>
</tr>
<tr>
<td>_ _</td>
<td>_ _</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PRICE</th>
<th>QTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>001 A</td>
<td>001 A</td>
</tr>
<tr>
<td>002 992</td>
<td>002 4</td>
</tr>
<tr>
<td>003 Price</td>
<td>003 Qty</td>
</tr>
<tr>
<td>004 D;PROD.NO</td>
<td>004 D;PROD.NO</td>
</tr>
<tr>
<td>_ _</td>
<td>_ _</td>
</tr>
</tbody>
</table>

V/CONV - Conversion Code

The V/CONV attribute (attribute 7) contains processing codes that Ultimate RECALL uses to convert data to output format. Typical uses are for formatting date, time, or money values by adding hyphens, commas, slashes, periods, or dollar signs. For information on processing codes, see Chapter 5.

Multiple processing codes can be specified by separating each code with a value mark (ASCII 253).

<table>
<thead>
<tr>
<th>COMPANY.L</th>
</tr>
</thead>
<tbody>
<tr>
<td>001 A</td>
</tr>
<tr>
<td>002 1</td>
</tr>
<tr>
<td>003 Company Name</td>
</tr>
<tr>
<td>004</td>
</tr>
<tr>
<td>005</td>
</tr>
<tr>
<td>006</td>
</tr>
<tr>
<td>007 MCTJML(#20)</td>
</tr>
<tr>
<td>008</td>
</tr>
<tr>
<td>009 L</td>
</tr>
<tr>
<td>010 20</td>
</tr>
</tbody>
</table>

V/CORR - Correlative Code

The V/CORR attribute (attribute 8) contains processing codes that Ultimate RECALL uses to convert data from its internal (stored) format to processing format. Typical uses are for defining attribute values that are calculated by arithmetic functions. The arithmetic can be performed using the stored value, or other specified attribute values. For information on processing codes, see Chapter 5.
Multiple codes can be specified by separating each code by a value mark (ASCII 253).

**V/TYP - Justification**

The V/TYP attribute (attribute 9) defines both the justification of the output (left- or right-justified) in columnar mode and the type of sort used when the file is sorted by this attribute.

Table 4-3 lists the valid V/TYP codes, and describes how data is placed and sorted. For additional information on sorting, see the subsection, Effects of V/TYP on Sorting, later in this chapter.

**V/MAX - Column Width**

The V/MAX attribute (attribute 10) defines the number of characters to display on a line (column width). If the column heading has more characters than specified in the V/MAX value, the column heading determines the column width.

If the attribute definition item has a V/MAX of 0 and an S/NAME of a backslash (\), the attribute is not displayed on detail lines.

The V/TYP code determines how the extra characters are displayed if the number of characters in the attribute exceeds the column width; see Table 4-3.

In order to ensure the proper display of data, this attribute should always be defined.
## Components of Attribute Definition Items

### Table 4-3. V/TYP Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Placement of Text</th>
<th>Sorting Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>left justifies data. If the number of characters in the attribute exceeds the column width, the extra characters are wrapped to the next line.</td>
<td>alphanumeric (ASCII) sort.</td>
</tr>
<tr>
<td>R</td>
<td>right justifies data. If number of characters in the attribute exceeds the column width, the extra characters overwrite columns to the left, up to the left margin.</td>
<td>numeric sort, when possible. This option allows for both alpha and numeric portions within a value. The numeric portions are sorted in the lowest-to-highest number order, while any non-numeric portions are sorted in ASCII code order (left-justified).</td>
</tr>
<tr>
<td>RN</td>
<td>right justifies data. If number of characters in the attribute exceeds the column width, the extra characters overwrite columns to the left, up to the left margin.</td>
<td>numeric sort only. This option is designed for all-numeric value fields, and treats decimal points and positive and negative signs (+ and -) as qualifiers of the numeric value. All other non-numeric characters in a value are treated as errors, and are sorted alphanumerically, before the numeric values.</td>
</tr>
<tr>
<td>T</td>
<td>left justifies data. If number of characters in the attribute exceeds the column width, the line is folded after the last blank preceding the end of the column.</td>
<td>alphanumeric (ASCII) sort.</td>
</tr>
<tr>
<td>U</td>
<td>left justifies data. Ignores V/MAX; text continues to end of attribute, wraps at end of screen. Used primarily for last attribute on line.</td>
<td>alphanumeric (ASCII) sort.</td>
</tr>
</tbody>
</table>
Attribute Definition Items

Effects of V/TYP on Sorting

A single data attribute often has several attribute definition items since different formatting, sorting, or selection requirements may require them. Each attribute definition item requires a different item-ID (such as NAME, EMP.NAME, FULLNAME, and EMPLOYEE LAST.NAME).

For example, multiple definition items can specify different V/TYP codes in order to sort an attribute in different ways. A left-justified V/TYP (L, T, or U) causes an alphanumeric sort (left to right, one character at a time, smallest to largest ASCII value). A right-justified V/TYP (R) causes a numeric sort for numeric characters and an alphanumeric sort for all other characters. A right-justified numeric V/TYP (RN) expects only numeric data; attributes with non-numeric characters are sorted alphabetically before attributes that are all numeric. Attributes that are all numeric are then sorted numerically.

The example in Figure 4-1 shows three attribute definition items for the same attribute. Each definition item causes a different sort to be used. The differences in the results of the sort on sample data are shown in Figure 4-2.

<table>
<thead>
<tr>
<th>item-ID</th>
<th>SORT.LEFT</th>
<th>SORT.RIGHT</th>
<th>SORT.RIGHTN</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>002</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>003</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>004</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>005</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>006</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>007</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>008</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>009</td>
<td>L</td>
<td>R</td>
<td>RN</td>
</tr>
<tr>
<td>010</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 4-1. Multiple Attribute Definition Items
### Components of Attribute Definition Items

<table>
<thead>
<tr>
<th>Data</th>
<th>SORT.LEFT</th>
<th>SORT.RIGHT</th>
<th>SORT.RIGHTN</th>
</tr>
</thead>
<tbody>
<tr>
<td>-123</td>
<td>-123</td>
<td>-123</td>
<td>100A</td>
</tr>
<tr>
<td>123.12</td>
<td>-123.12</td>
<td>-123.12</td>
<td>10B</td>
</tr>
<tr>
<td>-123.12</td>
<td>100A</td>
<td>1C</td>
<td>1C</td>
</tr>
<tr>
<td>20</td>
<td>10B</td>
<td>2</td>
<td>C1</td>
</tr>
<tr>
<td>2</td>
<td>123</td>
<td>10B</td>
<td>-123.12</td>
</tr>
<tr>
<td>10B</td>
<td>123.12</td>
<td>20</td>
<td>-123</td>
</tr>
<tr>
<td>C1</td>
<td>1C</td>
<td>100A</td>
<td>2</td>
</tr>
<tr>
<td>100A</td>
<td>2</td>
<td>123</td>
<td>20</td>
</tr>
<tr>
<td>1C</td>
<td>20</td>
<td>123.12</td>
<td>123</td>
</tr>
<tr>
<td>123</td>
<td>C1</td>
<td>C1</td>
<td>123.12</td>
</tr>
</tbody>
</table>

Figure 4-2. Effects of V/TYP on Sorting
5 Processing Codes

Processing codes convert data from internal (stored) format to intermediate or output format and are specified in attributes 7 or 8 of attribute definition items or file definition items. Processing codes in attribute 7 are called conversions; processing codes in attribute 8 are called correlatives.

A processing code consists of a letter followed by zero or more parameters. Table 5-1 lists the processing codes and their functions.

A processing code is applied whenever the attribute specified in the attribute definition item is not null. To ensure that a processing code that references multiple attributes is always applied, set the A/AMC of the attribute definition item to 0. This specifies the item-ID, which guarantees that the attribute is not null.

A processing code can be specified as either a conversion or correlative, depending on when it is desired that the special processing be applied to the data. For more information, see the section, Conversions and Correlatives, later in this chapter.

Ultimate RECALL provides some error checking for correct syntax in processing codes, but in many cases, if the syntax is not valid, no message is produced and null values are returned. If a processing code does not act as you expect, it may be because of an error in writing it. It is recommended that you read closely the description of each processing code.
Table 5-1. Processing Codes (1 of 2)

<table>
<thead>
<tr>
<th>Code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>arithmetic</td>
<td>computes mathematical and IF expressions; converted to an F code at run-time.</td>
</tr>
<tr>
<td>B</td>
<td>BASIC</td>
<td>calls BASIC subroutine.</td>
</tr>
<tr>
<td>C</td>
<td>concatenate</td>
<td>concatenates (chain together) attribute values.</td>
</tr>
<tr>
<td>D</td>
<td>date</td>
<td>converts date to external format.</td>
</tr>
<tr>
<td>F</td>
<td>function</td>
<td>computes mathematical functions.</td>
</tr>
<tr>
<td>G</td>
<td>group</td>
<td>extracts one or more fields separated by a specified delimiter.</td>
</tr>
<tr>
<td>L</td>
<td>length</td>
<td>tests attribute for length.</td>
</tr>
<tr>
<td>MCx</td>
<td>mask character</td>
<td>converts strings as specified by x.</td>
</tr>
<tr>
<td>ML</td>
<td>mask decimal,</td>
<td>formats and scales numbers and dollar amounts.</td>
</tr>
<tr>
<td></td>
<td>left justified</td>
<td></td>
</tr>
<tr>
<td>MP</td>
<td>mask packed</td>
<td>converts data values to or from packed decimal format.</td>
</tr>
<tr>
<td></td>
<td>decimal</td>
<td></td>
</tr>
<tr>
<td>MR</td>
<td>mask decimal,</td>
<td>formats and scales numbers and dollar amounts.</td>
</tr>
<tr>
<td></td>
<td>right justified</td>
<td></td>
</tr>
<tr>
<td>MD</td>
<td>mask time</td>
<td>converts time of day to external format.</td>
</tr>
<tr>
<td>MT</td>
<td>mask time</td>
<td></td>
</tr>
<tr>
<td>MX</td>
<td>mask hexadecimal</td>
<td>converts ASCII character strings to their hexadecimal (base 16) equivalents.</td>
</tr>
<tr>
<td>P</td>
<td>pattern match</td>
<td>tests attributes for those data values that match a specified pattern.</td>
</tr>
<tr>
<td>R</td>
<td>range</td>
<td>returns values that fall within a specified range.</td>
</tr>
<tr>
<td>S</td>
<td>substitution</td>
<td>substitutes values.</td>
</tr>
<tr>
<td>T</td>
<td>text extraction</td>
<td>extracts a fixed field from an attribute value.</td>
</tr>
</tbody>
</table>
### Table 5-1. Processing Codes (2 of 2)

<table>
<thead>
<tr>
<th>Code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tfile</td>
<td>file translation</td>
<td>verifies or converts attribute values by translating them through a specified file.</td>
</tr>
<tr>
<td>U</td>
<td>user-defined</td>
<td>invokes user-written routines.</td>
</tr>
<tr>
<td>V</td>
<td>item-ID extraction</td>
<td>specifies attribute that contains item-IDs for WITHIN connective.</td>
</tr>
</tbody>
</table>
Correlatives and Conversions

There are three formats of attribute data used by Ultimate RECALL:

<table>
<thead>
<tr>
<th>Format Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>internal</td>
<td>format of data when retrieved from storage; the stored format.</td>
</tr>
<tr>
<td>intermediate</td>
<td>format created by Ultimate RECALL for processing; the processing format.</td>
</tr>
<tr>
<td>external</td>
<td>format created by Ultimate RECALL for printing or displaying; the output format.</td>
</tr>
</tbody>
</table>

Correlatives are applied to the stored data to change it to the intermediate format. Conversions are applied to the intermediate format to change it to the external format. If there are no correlatives for an attribute, the intermediate format is the same as the internal format. Likewise, if there are no conversions, the external format is the same as the intermediate format.

The phrase output conversion refers to changing data from stored format to intermediate format using any correlatives, and then from intermediate format to output format using any conversions. The phrase input conversion refers to changing values in the Ultimate RECALL statement from external format to intermediate format using any conversions.

Input conversion is performed only on literal values enclosed in quotes in the Ultimate RECALL statement. The value must be entered as one of the following parts of the statement:

- selection criteria (WITH connective).
- print limiter.
- EXPlode limiter.

The intermediate format is used for most Ultimate RECALL processing, including sorting, matching selection criteria, and accumulating totals. It is considered the processing format because both stored and user-supplied values are converted to it before most processing takes place. That is, if the attribute definition item has a correlative code, the value stored in the attribute is converted before using it. If an attribute
Multiple processing codes can be defined in both conversions and correlatives. Each code is separated by a value mark (ASCII 253). Multiple codes are processed from left to right. For example, the following defines two codes: a B code, which calls a BASIC subroutine, followed by an MR code, which descales the value produced by the subroutine and formats it as dollars:

```
B:RUNNING.TOTAL]MR2S, (#10)
```
Processing Codes

A - Arithmetic Code

The A code derives the attribute value by a mathematical or conditional (IF) expression.

Syntax

A{;}operand{((proc.code))}operator{operand...}

; semicolon; allows compatibility with other codes, but is ignored if present in an A code.

operands can be any of the following; any operand can be preceded by a minus sign (-) to change the sign of the resulting value:

amc{R} attribute number. An amc of zero (0) specifies the item-ID. An amc can be followed by an R, which specifies repetition of the value (for example, 2R*3 specifies that attribute 2 be used repetitively to multiply each value in attribute 3).

N(name) attribute name that exists in the dictionary of the current file.

'literal' literal string; must be enclosed in single ('') or double quotes (".

D current system date in internal format.

LPV load previous value; should only be specified as the first operand of conversion code that is converting an intermediate value.

NB current control break level number. On a non-break line, NB has a value of 0. On a break line, the lowest level break is 1. On a grand-total line, NB has a value of 127.

ND detail-line counter; on detail lines, this counter has a value of 1. On break lines, has value of the number of lines since last control break. This is used to generate averages in conjunction with control breaks. On a grand total line, ND equals the item counter (NI).
NI  current item counter (number of items listed or selected).

NS  subvalue counter (for columnar listing only).

NV  multivalue counter (for columnar listing only).

T   current system time in internal format.

(proc.code) any processing code is applied to the preceding operand; must be enclosed in parentheses.

operators operators can be arithmetic, relational, or special.

arithmetic each arithmetic operator requires two operands.

+  addition.
-  subtraction.
*  multiplication.
/  division; returns an integer result; thus 7 divided by 2 evaluates to 3, not 3.5.
:  concatenation; appends the value of the second operand to the value of the first operand.

relational each relational operator requires two operands. A 1 (true) is returned if the stated relationship is true; otherwise, a 0 (false) is returned.

<  less than.
>  greater than.
<= less than or equal to.
>= greater than or equal to.
=  equal to.
#  not equal to.

special each special operator takes operands as specified.

R(operand1, operand2)  remainder function: returns the remainder of the first operand divided by the second.

S(operand)  summation function (for multivalued
attributes); sums the multiple values of the operand, if any.

**operand[start-char,no-chars]**
substring function; extracts a substring from an operand (same as BASIC substring); **start-char** is the starting character of the value; **no-chars** is the number of characters to extract; both **start-char** and **no-chars** can be a literal number in quotes, an attribute number, or an entire expression.

**IF expr1 THEN expr2 ELSE expr3**
conditional expression function; determines a value based on an IF-THEN-ELSE expression where **expr1**, **expr2**, and **expr3** are expressions and can contain attribute numbers, attribute names, numeric and string constants, and related operators (see above). Every IF expression must resolve to one value. If **expr1** is true, then **expr2** is returned by the **IF** function. If **expr1** is false, then **expr3** is returned.

Since an IF-THEN-ELSE expression is an expression, it can contain nested IF-THEN-ELSEs within it.

**Description**
Ultimate RECALL examines an A code and converts it into its equivalent F code format. Then it retrieves the specified attribute, converts the values if specified, and sets up the literals in the specified order. Finally, any arithmetic and comparison operations are performed and the value of the attribute is derived.

If the **N(name)** operand is specified, the dictionary is searched for the attribute **name**. If not found, an error message is printed. If found, the AMC (attribute 2) of that attribute definition item is used as the AMC in the A code. If the **N(name)** operand is specified in a conversion (attribute 7), any conversions in the attribute **name** are applied to derive the value. If the **N(name)** operand is specified in a correlative (attribute 8), any correlatives in the attribute **name** are applied to derive the value.
Note:  If the N(name) attribute is to be used with an arithmetic operator, be sure that the attribute definition item used does not include formatting values.

The operators and operands can be enclosed in parentheses; codes within parentheses are processed first. In the following example, the values of attribute 1 and 2 are added, and the resulting value is multiplied by the value of the attribute defined as COST:

A(1+2)*N(COST)

In general, blanks (spaces) can be used freely for easier reading of complex A codes. The following are equivalent:

A(1+2)*N(COST)
A (1 + 2) * N(COST)

An A code cannot refer back to itself. If the system encounters an A code that refers back to itself, the following message is displayed:


attribute name

Order of Operations

Expressions are evaluated based on their operators, in the following order (first to last):

1  ()
2  *  /
3  +  -
4  <  >  <=  >=  =  \

Two operators with the same precedence are evaluated from left to right. Expressions inside parentheses are evaluated first. If nested IF expressions are found, each nested IF-THEN-ELSE expression is evaluated as if it were within parentheses. The resulting value from the A code calculation is then available for processing the statement or to fill in the current item's detail line of output.

| A·2·3<4 | is equivalent to | ((1·2)+3)<4 |
|---------|------------------|
| 4>=5-2/1 | is equivalent to | 4<=(5-(2/1)) |
Processing Codes

\[
\begin{align*}
1+2-3 & \quad \text{is equivalent to} \quad (1+2)-3 \\
4/5*6 & \quad \text{is equivalent to} \quad (4/5)*6
\end{align*}
\]

\[
\begin{align*}
\text{A IF 6['1','1'] = '*' THEN IF 2 > "10" THEN 'OK' ELSE ' ELSE '}
\end{align*}
\]

\[
\begin{align*}
\text{is equivalent to:} \quad \\
\text{A IF 6['1','1'] = '*' THEN (IF 2 > "10" THEN 'OK' ELSE ' ELSE ') ELSE '}
\end{align*}
\]

Totals

When producing totals (that is, using the TOTAL modifier), an A code used as a correlative (attribute 8) is processed differently from an A code used as a conversion (attribute 7).

- an A code used as a correlative is applied to attribute values on detail lines. The correlated value is accumulated into the total to be output on the break line (subtotals). The A correlative is then ignored on break lines and total lines. If the correlated value has non-numeric characters in it, then only the digits, if any, to the left of the first non-numeric character are totaled.

- an A code used as a conversion is applied to the total on break lines (subtotals) and total lines as well as being applied to attribute values on detail lines. The value on the detail line is accumulated into the total only if the AMC in attribute 2 of the attribute definition item is real, or if the attribute number (AMC) operand in the conversion matches an attribute being totaled in the statement (specified with a TOTAL modifier). If such an attribute is not in the statement, a value of zero is returned as the total for this attribute. If the value on the detail line is accumulated into the total, it is added before the conversion is applied.

You can use A processing codes for both correlatives and conversions in a single attribute definition item to produce correctly formatted values on the report. In order to use a correlated value in the conversion, use LPV as the first operator in order to reload the A correlative results on to the stack.

Dummy attributes are often set up for totaling purposes, since an attribute referenced in an A conversion (attribute 7) must be totaled in order to be meaningful on a break line.
The ND counter keeps track of the number of detail lines within each control break. This counter can be useful to compute average values for attributes and report the averages on break lines and grand-total lines.

```plaintext
:CT DICT INVOICE AVERAGE
  Average
  001 A
  002 0
  003 Average Owed
  004
  005
  006
  007 A:LPV/NDV/VRF(1)
  008 A4:ICTPRCLL/LS/PL
  009 H
  010 12

:SORT INVOICE TOTAL AVERAGE DET-SUPP

PAGE 1

INVOICE... Average Owed

***

6 items listed.
```
## Processing Codes

<table>
<thead>
<tr>
<th>Processing Code</th>
<th>Data in Attribute</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1+2</td>
<td>001 213</td>
<td>416</td>
</tr>
<tr>
<td></td>
<td>002 203</td>
<td></td>
</tr>
<tr>
<td>A&quot;10&quot;*3</td>
<td>003 96</td>
<td>960</td>
</tr>
<tr>
<td>AS(5&quot;100&quot;)</td>
<td>005 1]2]3</td>
<td>306</td>
</tr>
<tr>
<td>A; (NI)/&quot;43&quot;</td>
<td>If NI = 42</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>If NI = 44</td>
<td>1</td>
</tr>
<tr>
<td>A;2R * 3</td>
<td>002 2</td>
<td>21416</td>
</tr>
<tr>
<td></td>
<td>003 1]2]3</td>
<td></td>
</tr>
<tr>
<td>A:2 * 3</td>
<td>002 2</td>
<td>21010</td>
</tr>
<tr>
<td></td>
<td>003 1]2]3</td>
<td></td>
</tr>
<tr>
<td>A:IF 1 &gt; &quot;100&quot; AND 2 &lt; 3 THEN &quot;EUREKA&quot; ELSE &quot;&quot;</td>
<td>001 101</td>
<td>EUREKA</td>
</tr>
<tr>
<td></td>
<td>002 90</td>
<td></td>
</tr>
<tr>
<td></td>
<td>003 95</td>
<td></td>
</tr>
<tr>
<td>A IF 9(LO) = '9' THEN 9(ML#5-#4) ELSE 9(ML#5)</td>
<td>009 92006</td>
<td>92006</td>
</tr>
<tr>
<td></td>
<td>009 920061234</td>
<td>92006-1234</td>
</tr>
<tr>
<td>A:IF 1 &gt; '100' THEN &quot;<em>REORDER</em>&quot; ELSE &quot;&quot;</td>
<td>001 254</td>
<td><em>REORDER</em></td>
</tr>
<tr>
<td>A1['1', '999'*(2=3)]</td>
<td>001 12334567890</td>
<td>1234567890</td>
</tr>
<tr>
<td></td>
<td>002 350</td>
<td></td>
</tr>
<tr>
<td></td>
<td>003 350</td>
<td></td>
</tr>
<tr>
<td></td>
<td>001 12334567890</td>
<td>(null)</td>
</tr>
<tr>
<td></td>
<td>002 350</td>
<td></td>
</tr>
<tr>
<td></td>
<td>003 351</td>
<td></td>
</tr>
<tr>
<td>A(N(PRICE) *N(QTY)) (MR2$) (where PRICE is attribute 5 and QTY is attribute 6)</td>
<td>005 255013500</td>
<td>$25500.00</td>
</tr>
<tr>
<td></td>
<td>006 10001000</td>
<td>$35000.00</td>
</tr>
</tbody>
</table>
B - BASIC Subroutine Call Code

The B code calls a BASIC subroutine to process an attribute's value on input or before output.

Syntax

```
B;{filename }subrname
;
```

- `filename`: the file in which the subroutine resides. If `filename` is specified, the subroutine need not be cataloged; if `filename` is not present, the subroutine must be cataloged.
- `subrname`: item-ID of the subroutine; must be cataloged in the MD of any account that specifies the attribute definition item.

Description

The B code can be used in situations where none of the other processing codes can manipulate the stored data in the manner desired.

An item in the SYSLIB file called RCL.COMMON must be INCLUDEd in the subroutine called by the B processing code to give the subroutine access to the `filename`, current item-ID, and other information; see Table 5-2 for a list of the variables in RCL.COMMON. The RCL.COMMON item also sets the PRECISION to zero.

This item should be INCLUDEd in the subroutine as follows:

```
001 SUBROUTINE name
002 $INCLUDE SYSLIB RCL.COMMON
```

Most RCL.COMMON variables are read-only and do not affect Ultimate RECALL operations, even if the BASIC subroutine changes them. A Y in the READ-only column in Table 5-2 indicates the variables that are read-only.

When the subroutine is called from Ultimate RECALL, its local variables are initialized only on the first call, where they are set to null and are not flagged as uninitialized variables. This is unlike subroutine calls from BASIC, where local variables are initialized on each entry to the subroutine. Thus, you can use any local variables in the subroutine to store data such as totals, or to open files. The variables retain the stored data until the Ultimate RECALL statement is completed.
If the subroutine must write to a file, the Ultimate RECALL statement must include the W option.

**C.DATA**

The C.DATA variable from the RCL.COMMON subroutine can be used by the subroutine to store data to be used as the value of the attribute. For example, the data can be displayed via a LIST-type statement, selected by an output specification in a SELECT statement, or sorted.

The C.DATA variable can contain more than one value separated by value marks. However, multiple subvalues in C.DATA are ignored; that is, only the first subvalue is displayed or stored.

**Reserved BASIC Operations**

The following BASIC operations are not allowed in subroutines called from Ultimate RECALL:

- file writes, deletes, or clears to the file variables C.PRIMFILE, C.DICTPRIMFILE, or to the files they refer to.
- file writes, deletes, or clears to any other file opened by the subroutine, unless the Ultimate RECALL statement has the W option.
- any tape operations, such as WRITET.
- EXECUTE, CHAIN, and ENTER statements.
- GET function.

In addition, the BASIC system variable @SENTENCE and the equivalent system call SYSTEM(18), when used in a subroutine called by Ultimate RECALL, return internal Ultimate RECALL information, not the TCL statement.
## Table 5-2. RCL.COMMON Variables

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>READ-only</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.DICTPRIMFILE</td>
<td>Y</td>
<td>file variable for DICT of file.</td>
</tr>
<tr>
<td>C.PRIMFILE</td>
<td>Y</td>
<td>file variable for data section.</td>
</tr>
<tr>
<td>C.ITEMID</td>
<td>Y</td>
<td>current item-ID.</td>
</tr>
<tr>
<td>C.ITEM</td>
<td>Y</td>
<td>contents of current item.</td>
</tr>
<tr>
<td>C.AMC</td>
<td>Y</td>
<td>attribute number of attribute currently being processed.</td>
</tr>
<tr>
<td>C.VMC</td>
<td>Y</td>
<td>value number of value currently being processed.</td>
</tr>
<tr>
<td>C.SVMC</td>
<td>Y</td>
<td>subvalue number of subvalue currently being processed.</td>
</tr>
<tr>
<td>C.FLAGS</td>
<td>N</td>
<td>flags; see Table 5-3.</td>
</tr>
<tr>
<td>C.PRIMFILENAME</td>
<td>Y</td>
<td>name of the current file.</td>
</tr>
<tr>
<td>C.USER(15)</td>
<td>N</td>
<td>current data value.</td>
</tr>
</tbody>
</table>

## Table 5-3. The C.FLAGS Variable

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>processing a detail line.</td>
</tr>
<tr>
<td>1-15</td>
<td>processing a BREAK line (subtotal line); the C.FLAGS value is the break level number.</td>
</tr>
<tr>
<td>127</td>
<td>processing GRAND-TOTAL line.</td>
</tr>
<tr>
<td>-1</td>
<td>processing a sort or selection.</td>
</tr>
</tbody>
</table>
The subroutine in the following example performs a translation from a multivalued field of contact numbers stored in attribute 1 of the CONTACTS.X file to obtain multivalued phone numbers from attribute 7 of the CONTACTS file. Note that each multivalue in the CONTACTS.X file can itself retrieve multiple phone numbers.

Attribute definition item for the PHONE attribute:
001 A
002 I
...
007 ML(###/### ###)
008 B:TRANS.PHONES
009 L
010 15

Attribute definition item for the CONTACT# attribute:
001 A
002 I
...
009 L
010 15

Subroutine TRANS.PHONES:
001 SUBROUTINE TRANS.PHONES
002 INCLUDE SYSLIB MCL.COMMON
003 IF FIRST ELSE
004 FIRST = 1 * first line only once in report
005 OPEN 'CONTACTS.X' FOR INPUT FIRST IO CONTACTS
006 END
007 READ C.DATA NOW C1, C2, C3, C4, C5, C6, C7, C8, C9, C10...
008 RETURN.

The READY in the subroutine picks up multivalued phone numbers from the CONTACTS file. These phone numbers are returned in the C.DATA variable. Ultimate RECALL displays them as follows:

```
LIST CONTACTS.X 'ULT' CONTACT# PHONE
```

End of list.
The subroutine in the following example generates a running total.

```
Attribute definition item for the CUMULATIVE attribute in the INVOICE file; it will display only on break lines:

   CUMULATIVE
001  A
002  90
003  '\n
007  B:RUNNING.TOTAL: PAGE, (*:1)
008  A4+3:PROD.NO:X/XX:
009  L
010  O

Subroutine RUNNING.TOTAL:
001  SUBROUTINE RUNNING.TOTAL
002  INCLUDE SYSLIB ROL.COMMON
003  IF C.FLAGS > 0 THEN
004   IF BREAK THEN = return the running total value
005   ELSE DATA TOTAL:
006  END ELSE
007  IF NOT(CUMULATIVE) THEN TOTAL C. DATA
008  END DATA CUMULATIVE
009  END
010  RETURN

Ultimate RECALL displays the running total as follows:

:SORT INVOICE BY ZIP BREAK-ON ZIP.L "L" TOTAL TOT.PRICE
CUMULATIVE
PAGE     Date: 1 JAN 1992

INVOICE... ...  ... ...
1683    ...    14.80
        ...    ...
1644    ...    18.21
        ...    ...
1694    ...    18.32
        ...    ...
1681    ...    17.14
        ...    ...
1691    ...    14.36
        ...    ...
163    ...    ...
        ...
...  ...
6 ITEMS TOTAL.
```
The C code concatenates attributes and literal values.

Syntax

\[
\text{C\{;}\text{val1}\{x1\{val2\{x2...xn-1\{valn}}}}
\]

- **semicolon**: allows compatibility with other codes, but is ignored if present in a C code.
- **val1**: first value to use. It can be an attribute number (AMC); any literal string enclosed in single quotes ('), double quotes ("), or backslashes (\); or an asterisk (*), which specifies using the last value generated from a conversion or correlative operation.
- **x1**: separation character to be inserted after the first value. It can be any non-numeric character (including a blank) except a system delimiter (segment mark, attribute mark, value mark, or subvalue mark). A semicolon (;) specifies that no separation character is to be inserted.
- **val2 thru valn**: second to nth values to concatenate; each has the same criteria as val1.
- **x2 thru xn-1**: second to nth-1 separation character; each has the same criteria as val1.

Description

The C code can be used whenever fields need to be concatenated.

<table>
<thead>
<tr>
<th>Code</th>
<th>Attribute Value</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2; 'ANS'=3+1</td>
<td>001 42 002 THE 003 1000</td>
<td>THE ANS=1000+42</td>
</tr>
<tr>
<td>C&quot;NAME&quot;:4,5</td>
<td>004 STERN 005 FRANK</td>
<td>NAME:STERN,FRANK</td>
</tr>
<tr>
<td>C*%6</td>
<td>006 INTEREST last computed value: 12.73</td>
<td>12.73 INTEREST</td>
</tr>
<tr>
<td>C9;10</td>
<td>009 PT# 010 203</td>
<td>PT#203</td>
</tr>
</tbody>
</table>
D - Date Code

The D code converts a date to or from its internal format. It is generally used as a conversion.

Syntax

\[ D\{n\}{*m}\{s\} \]

- **n**: number of year digits to output; could be 0, 1, 2, 3, or 4. If omitted, all four year digits are displayed. If 0 is specified, no year is displayed.
- **m**: used for group extraction of a date from an attribute. If omitted, the date is assumed to comprise the entire attribute value.
  - *represents the field delimiter character in the attribute; it can be any non-numeric character (including a space), except a semicolon (;) or a system delimiter (segment mark, attribute mark, value mark, or subvalue mark).
- **m**: specifies the number of fields to skip (before the date field).

For more information on group extraction, see the G processing code.

- **s**: can be either a non-numeric character to be used as a date separator, or a special subcode.

  - If **s** is a date separator character, it also specifies an all numeric output date format (for example: 12/13/92). If no **s** parameter is present, the output date format is an alphabetic month with a numeric day and year (for example: 13 DEC 1992).

  - If **s** is a special subcode, it can be one of the following:
    - **D**: displays only the 2-digit number of the day of the month.
    - **M**: 2-digit number of the month.
    - **Q**: 1-digit quarter.
    - **J**: 5-digit Julian date.
    - **Y**: year; number of digits depends on **n**.
Processing Codes

Description

The D code directs Ultimate RECALL to convert the date value from its internal format. In internal format, dates are stored as the integer number of days between that date and the zero date, December 31, 1967. Dates before 12/31/67 are stored as negative numbers; dates after 12/31/67 are positive numbers.

In order for dates to sort in correct date order, they should be stored in internal format, not as days, months, and years, and the date processing code should be specified as a conversion. (BASIC can convert dates using the ICONV and OCONV functions.)

Also, because the internal format is numeric, dates should be specified as right-justified. They may not be compared correctly if the attribute is left-justified. For example, dates on or after May 18, 1995 have a five-digit internal date, starting at 10000. With a left-justified attribute, these dates will sort before dates such as May 17, 1995, which has an internal format of 9999. If the attribute is defined as right-justified, the dates are sorted correctly.

If the value of the specified attribute is a valid internal date, Ultimate RECALL converts the date to its output format, as specified in the D code, and the result is used in the output. If the value of the specified attribute is not a valid internal date, the original value is returned.

If an attribute with a D code is used in an Ultimate RECALL statement, and a date is specified in the statement, that date will be converted into internal format for sort and select comparisons.

Note: When dates are specified for input conversion, if the year is not included, the current system year is assumed. If a 2-digit year is input as 30-99, it is assumed to be 1930-1999; if a 2-digit year is 00-29, it is assumed to be 2000-2029.

The following shows internal and external date formats:
## D - Date Code

<table>
<thead>
<tr>
<th>Internal (Stored) Format</th>
<th>External (Output) Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>-100</td>
<td>22 SEP 1967</td>
</tr>
<tr>
<td>-10</td>
<td>21 DEC 1967</td>
</tr>
<tr>
<td>0</td>
<td>31 DEC 1967</td>
</tr>
<tr>
<td>1</td>
<td>01 JAN 1967</td>
</tr>
<tr>
<td>100</td>
<td>09 APR 1968</td>
</tr>
<tr>
<td>1000</td>
<td>25 SEP 1970</td>
</tr>
<tr>
<td>10000</td>
<td>18 MAY 1995</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>Attribute Value</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>8888</td>
<td>01 MAY 1992</td>
</tr>
<tr>
<td>D2</td>
<td>8888</td>
<td>01 MAY 92</td>
</tr>
<tr>
<td>D4</td>
<td>8888</td>
<td>01 MAY 1992</td>
</tr>
<tr>
<td>D0</td>
<td>8888</td>
<td>01 MAY</td>
</tr>
<tr>
<td>D1</td>
<td>8888</td>
<td>05-01-1992</td>
</tr>
<tr>
<td>D2/</td>
<td>8888</td>
<td>05/01/92</td>
</tr>
<tr>
<td>D2*</td>
<td>8888</td>
<td>05-01-92</td>
</tr>
<tr>
<td>D*</td>
<td>8888</td>
<td>05/01/1992</td>
</tr>
<tr>
<td>D-1</td>
<td>INV^8888</td>
<td>INV-01 MAY 1992</td>
</tr>
<tr>
<td>DY</td>
<td>8888</td>
<td>1992</td>
</tr>
<tr>
<td>D2Y</td>
<td>8888</td>
<td>92</td>
</tr>
<tr>
<td>DQ</td>
<td>8888</td>
<td>2 (second quarter)</td>
</tr>
<tr>
<td>DD</td>
<td>8888</td>
<td>1 (first day)</td>
</tr>
<tr>
<td>DM</td>
<td>8888</td>
<td>5 (fifth month)</td>
</tr>
<tr>
<td>DJ</td>
<td>8888</td>
<td>122 (Julian date)</td>
</tr>
</tbody>
</table>
The F code performs a mathematical or manipulative function on one or more operands.

Syntax

F{ ; }element{ ; element{ ; element{ ; ...} }}

;  semicolon, required separator between elements. A semicolon immediately following the F code is ignored.

element  can be either an operand or an operator. F code elements can be made up of any number of operators and operands, each separated by a semicolon (;).

operand  specifies the value to be pushed onto the stack; can be any one of the following:

AMC    number of attribute to be pushed on to stack; the attribute can be multivalued.

AMC(processing code)  number of attribute followed by processing code enclosed in parentheses.

'literal'  string constant; can be any string enclosed in single or double quotes.

Cn     integer decimal number constant; the string cannot contain a comma (,).

D       current date in internal format.

LPV     load previous value: should only be specified as the first operand of an F conversion code that converts the intermediate value of an attribute.

NB      current control break level number. On a non-break line, NB has a value of 0. On a break line, the lowest level break is 1. On a grand-total line, NB has a value of 27.

ND      detail-line counter: on detail lines, this counter has a value of 1. On break lines, has value of the number of lines since last control break. This is used to generate averages in conjunction
with control breaks. On a grand total line, ND equals the item counter (NI).

NI current item counter (number of items listed or selected).

NS subvalue counter (for columnar listing only).

NV multivalue counter (for columnar listing only).

T current time in internal format.

R any of the operands can optionally be followed by an R, which specifies repetition of the value so that a single value can be used the same number of times as a multivalued attribute used in the expression.

operator an operator takes the current stack values, pops them off the stack, performs the specified operation, then returns the result to the stack as noted; operator can be any one of the following:

* { n } multiplies stack2 value by the stack1 value; the result is stored in stack1; if n is specified, product is descaled by n (result is divided by \(10^{n-1}\)).

/ divides stack2 value by the stack1 value; the quotient is stored in stack1 (integer value, with any fractional portion truncated).

R divides as above, but remainder is stored in stack1 instead of the quotient.

+ adds stack1 and stack2 values; the sum is stored in stack1.

- subtracts stack1 value from the stack2 value; the difference is stored in stack1.

: concatenates the string value in stack1 to the end of the string value in stack2. The concatenated string is stored in stack1.

[ ] extracts a subset of the string value in stack3, using the stack2 value as the starting character position and the stack1 value as the number of characters to extract. The extracted subset is stored in stack1.
S sums all values currently in stack 1 and stores the sum in stack 1.

Caution: Although the S operator sums only the values in stack 1, it may use other stack positions for its operations. For this reason, values placed on the stack before the S operator is called may not be valid after the S operator is finished.

_ (underscore) exchanges the values in stack 1 and stack 2.

P pushes the value in stack 1 back onto the stack; that is it duplicates the stack 1 value and stack 1 and stack 2 will each have the same value. The original stack values are pushed down one stack; no values are popped.

(processing code) processing code enclosed in parentheses. The code is applied to the stack 1 value; the converted value replaces the original value in stack 1. The code can include parentheses, as in pattern matching and output masks.

relational operators tests stack 1 and stack 2 values to determine if the stated relation is true or false. If true, a 1 (one) is stored in stack 1; if false, a 0 (zero) is stored in stack 1.

= equal-to: tests if stack 1 value is equal to stack 2 value.

< less-than: tests if stack 1 value is less than stack 2 value.

> greater-than: tests if stack 1 value is greater than stack 2 value.

# not-equal: tests if stack 1 value is not equal to stack 2 value.

[ less-than or equal-to: tests if stack 1 value is less than or equal to stack 2 value.

] equal-to or greater-than: tests if stack 1 value is equal or greater than stack 2 value.
F code functions are set up in reverse Polish notation, which means that operands are specified before the operator in the function code. This differs from algebraic notation, where the operator is specified between the operands; algebraic notation is used in A processing codes.

If an F-code operand is multivalued, the result is also multivalued. If the operator requires two operands, the result of the operation has as many values as the longer of the two operands. If the operator is an arithmetic function, zeros are substituted for null values in the shorter list; otherwise nulls are used. To repeat the use of the last value of an operand, specify the R option with that operand.

Note: Many F codes can be specified in a simpler format by using an A code, which will be automatically converted at run-time to the equivalent F code. For example, to divide the value in attribute 40 by the value in attribute 20, you could set up either an A or F code as follows; the results are equivalent:

F code format (reverse Polish notation) 40;20;1
A code format (algebraic notation) 40 / 20

F-Code Stack
The operand values in F codes are stored in a pushdown stack; the operator values direct Ultimate RECALL to perform the operation using the top stack (operand) values.

The F code pushes the operands on the stack until an operator is encountered. When an operand is pushed on to the stack, it is always pushed on to stack 1. The existing operands are moved down one stack entry. Up to 24 operands can be stacked.

When an operator is encountered, as many operands as are required by the operator are removed (popped) from the stack, starting with stack 1. The values are processed and the result pushed on to stack 1. This push-down, pop-up process repeats until the entire F code has been processed. Then the stack 1 value becomes the value of the F code correlative or conversion function.
Figure 5-1 shows the steps used internally to resolve the following F code:

\[ F; C3; C4; C20; +; * \]

<table>
<thead>
<tr>
<th></th>
<th>C3: push constant 3 onto stack</th>
<th>C4: push constant 4 onto stack</th>
<th>C20: push constant 20 onto stack</th>
<th>+: add stack1 and stack2</th>
<th>*: multiply stack2 by stack1</th>
</tr>
</thead>
<tbody>
<tr>
<td>stack1</td>
<td>3</td>
<td>4</td>
<td>20</td>
<td>24</td>
<td>72</td>
</tr>
<tr>
<td>stack2</td>
<td></td>
<td>3</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>stack3</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5-1. F-Code Stack Activity to Resolve

\[ F; C3; C4; C20; +; * \]
Figure 5-2 shows the stack activity of an F code that uses concatenation and substring operators to build a parts code.

TCL statement using CODE attribute definition item:

```
:LIST PARTS '29' CODE
```

Resulting listing:

```
PAGE 1 09:32:42 09 JUL 1992

PARTS..... CODE.....
29 800#29
```

End of list

Attribute definition item CODE in DICT PARTS file:

```
001 A
002 0
.
008 F;2;3;*:C#;:1;C3;C2;[];
009 L
010 10
```

Data item '29' in PARTS file

```
001 W/29
002 20
003 40
```

Stack activity as F code is being processed:

```

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>*</td>
<td>C#</td>
<td>:</td>
<td>1</td>
<td>C3</td>
<td>C2</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>stack1</td>
<td>20</td>
<td>40</td>
<td>800</td>
<td>#</td>
<td>800#</td>
<td>W/29</td>
<td>3</td>
</tr>
<tr>
<td>stack2</td>
<td>20</td>
<td>40</td>
<td>800</td>
<td>800#</td>
<td>W/29</td>
<td>3</td>
<td>800#</td>
</tr>
<tr>
<td>stack3</td>
<td>20</td>
<td>800</td>
<td>800#</td>
<td>W/29</td>
<td>3</td>
<td>800#</td>
<td></td>
</tr>
<tr>
<td>stack4</td>
<td>800#</td>
<td>W/29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stack5</td>
<td>800#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 5-2. F-Code Stack Activity to Resolve
F;2;3;*:C#;:1;C3;C2;[];:
When producing totals (that is, using the TOTAL modifier), an F code used as a correlative (attribute 8) is processed differently from an F code used as a conversion (attribute 7).

- An F code used as a correlative is applied to attribute values on detail lines. The correlated value is accumulated into the total to be output on the break line (subtotals). The F correlative is then ignored on break lines and total lines. If the correlated value has non-numeric characters in it, then only the digits, if any, to the left of the first non-numeric character are totaled.

- An F code used as a conversion is applied to the total on break lines (subtotals) and total lines as well as being applied to attribute values on detail lines. The value on the detail line is accumulated into the total only if the AMC in attribute 2 of the attribute definition item is real, or if the attribute number (AMC) operand in the conversion matches an attribute being totaled in the statement (specified with a TOTAL modifier). If such an attribute is not in the statement, a value of zero is returned as the total for this attribute. If the value on the detail line is accumulated into the total, it is added before the conversion is applied.

You can use F processing codes for both correlatives and conversions in a single attribute definition item to produce correctly formatted values on the report. In order to use a correlated value in the conversion, use LPV as the first operator in order to reload the F correlative results on to the stack.

Dummy attributes are often set up for totaling purposes, since an attribute referenced in an A conversion (attribute 7) must be totaled in order to be meaningful on a break line.
Averages

The ND counter keeps track of the number of detail lines within each control break. This counter can be useful to compute average values for attributes and report the averages on break lines and grand-total lines.

The following example is similar to the example shown with the A processing code, except it uses F correlatives and conversions.

```plaintext
:CT DICT INVOICE F.AVERAGE

F.AVERAGE
001 A
002 
003 Average Value
004
005
006
007 F.REPLACE "Avg" 
008 F.REPLACE "Total Avg"
009 
010 

:SORT INVOICE TOTAL F.AVERAGE DET-SUPP

PAGE 4 JUL 1992

INVOICE... DET-SUPP...

6 100-1 00-1
```
The following examples illustrate general uses for F processing codes.

<table>
<thead>
<tr>
<th>Code</th>
<th>Attribute</th>
<th>Value</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>F;1;2;*</td>
<td>001  20</td>
<td></td>
<td>800</td>
</tr>
<tr>
<td></td>
<td>002  40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F;1;2;*</td>
<td>001  20</td>
<td></td>
<td>800</td>
</tr>
<tr>
<td></td>
<td>002  40</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>003  30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F;1;2R;*</td>
<td>001  20</td>
<td></td>
<td>800</td>
</tr>
<tr>
<td></td>
<td>002  40</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>003  30</td>
<td></td>
<td>1200</td>
</tr>
<tr>
<td>F;1;S;2;*</td>
<td>001  20</td>
<td></td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td>002  40</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>003  30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F;1;3; (TPROD.NO:X;2;2);*</td>
<td>001  20</td>
<td></td>
<td>20800</td>
</tr>
<tr>
<td></td>
<td>002  40</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>003  7015</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROD.NO file, item 7015</td>
<td></td>
<td></td>
<td>002 1040</td>
</tr>
<tr>
<td>F;1; (ML#10);2;:</td>
<td>001  20</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>002  40</td>
<td></td>
<td>40</td>
</tr>
<tr>
<td>F:D; (DY);4;(DY):-</td>
<td>004  7900</td>
<td></td>
<td>3 (if current year is 1992)</td>
</tr>
</tbody>
</table>
G - Group Code

The G code extracts one or more contiguous fields from an attribute.

Syntax

\[ G\{m\}^*n \]

- \( m \): number of fields to skip. If omitted, zero is assumed and no fields are skipped.
- \( \ast \): character in the attribute used as the field delimiter. It can be any non-numeric character (including a space), except a system delimiter (segment mark, attribute mark, value mark, or subvalue mark).
- \( n \): the number of contiguous fields to extract.

Description

The G code is used when the stored format of data in an attribute is separated into delimited fields of information, and only some fields are required in processing the statement.

The specified number of fields (if any) are skipped, and the data in the specified number of fields is used. If the specified field delimiter is not found, the entire value is considered to be one field.

<table>
<thead>
<tr>
<th>Code</th>
<th>Attribute Value</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>G/1</td>
<td>05/01/92</td>
<td>05</td>
</tr>
<tr>
<td>GL/1</td>
<td>05/01/92</td>
<td>01</td>
</tr>
<tr>
<td>G2</td>
<td>01 MAY 1992</td>
<td>1992</td>
</tr>
<tr>
<td>G 2</td>
<td>01 MAY 1992</td>
<td>01 MAY</td>
</tr>
<tr>
<td>G1</td>
<td>01 MAY 1992</td>
<td>MAY 1992</td>
</tr>
<tr>
<td>G+1</td>
<td>123-455-789</td>
<td>123</td>
</tr>
<tr>
<td>G5+1</td>
<td>123-455-789</td>
<td>null</td>
</tr>
<tr>
<td>G1*1</td>
<td>&quot;START HERE&quot;</td>
<td>START HERE</td>
</tr>
<tr>
<td>G13</td>
<td>123-456789</td>
<td>456789</td>
</tr>
<tr>
<td>G\3</td>
<td>40#50#60#70#80</td>
<td>40#50#60#70#80#</td>
</tr>
</tbody>
</table>

Ultimate RECALL User's Guide
Confidential and Proprietary to The Ultimate Corp.
Processing Codes

L - Length Code

The L code tests an attribute value based on length (number of characters). If the value does not meet the criteria, a null value is returned.

Syntax

L n{,m}

n minimum number of characters in the attribute. If n is 0 or non-numeric, the number of characters in the attribute is returned. If n is numeric and greater than zero and m is not specified, the attribute must contain exactly n characters.

m specifies the maximum number of characters the attribute can contain.

Description

The L code is used when the stored format of data in an attribute needs to be tested for a valid number of characters before processing the statement, or to determine the number of characters in the attribute.

<table>
<thead>
<tr>
<th>Code</th>
<th>Attribute Value</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>L5</td>
<td>92705</td>
<td>92705</td>
</tr>
<tr>
<td>L9</td>
<td>92705</td>
<td>(null)</td>
</tr>
<tr>
<td>L5, 9</td>
<td>92705</td>
<td>92705</td>
</tr>
<tr>
<td>L6, 8</td>
<td>92705</td>
<td>(null)</td>
</tr>
<tr>
<td>L0</td>
<td>92705</td>
<td>5</td>
</tr>
</tbody>
</table>
MC - Mask Character Code

The MC code converts an attribute value according to a character code.

Syntax

\[ MCx \]

- **x** subcode that determines the type of character conversion or extraction to be performed; the valid forms of the code are

  - **MCA** extracts and prints all alphabetic letters (both upper and lower case); non-alphabetic characters are not extracted or displayed.
  - **MC/A** extracts and prints all non-alphabetic characters; alphabetic characters are not extracted nor displayed.
  - **MCD** converts a decimal value to its hexadecimal (base 16) equivalent.
  - **MCL** converts all upper case letters to lower case. Lower case and non-alphabetic characters are not affected.
  - **MCN** extracts and prints all numeric characters (0-9); non-numeric characters are not extracted or printed.
  - **MC/N** extracts and prints all non-numeric characters; numeric characters (0-9) are not extracted or printed.
  - **MCP** converts unprintable ASCII characters into periods (.). The unprintable range is X'00' to X'1F' and X'7F' to X'FB'. ASCII printable characters (characters between X'20' and X'7E') are not affected.
  - **MCT** converts each word of data to lower case with the initial letter capitalized (upper case). Non-alphabetic characters are not affected. (Characters separated by spaces or system delimiters are assumed to be words.)
  - **MCU** converts all lower case letters to upper case. Upper case and non-alphabetic characters are not affected.
  - **MCX** converts a hexadecimal (base 16) value to its decimal equivalent.

Description

The MC code can be used to convert an attribute value to upper or lower case, to extract either numeric or alphabetic characters, or to convert a
numeric value from decimal to hexadecimal or from hexadecimal to decimal.

For conversion of input data, the MC code directs Ultimate RECALL to convert the input value according to the subcode, whenever possible, to enable matching to the stored value.

<table>
<thead>
<tr>
<th>Code</th>
<th>Attribute Value</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCA</td>
<td>Alpha and 1234</td>
<td>Alpha and 1234</td>
</tr>
<tr>
<td>MC/A</td>
<td>Alpha and 1234</td>
<td>1234 (two spaces precede the numbers)</td>
</tr>
<tr>
<td>MCN</td>
<td>Alpha and 1234</td>
<td>1234</td>
</tr>
<tr>
<td>MC/N</td>
<td>Alpha and 1234</td>
<td>Alpha and</td>
</tr>
<tr>
<td>MCL</td>
<td>Alpha and 1234</td>
<td>Alpha and 1234</td>
</tr>
<tr>
<td>MCU</td>
<td>Alpha and 1234</td>
<td>ALPHA AND 1234</td>
</tr>
<tr>
<td>MCT</td>
<td>Alpha and 1234</td>
<td>Alpha And 1234</td>
</tr>
<tr>
<td>MCD</td>
<td>1234</td>
<td>4D2</td>
</tr>
<tr>
<td>MCX</td>
<td>1234</td>
<td>4660</td>
</tr>
<tr>
<td>MCP</td>
<td>X'0041FB' (3 bytes)</td>
<td>.A</td>
</tr>
</tbody>
</table>
MD - Mask Decimal Code

The system automatically converts an MD code into an MR (mask decimal, right justify) code. For more information, see the MR code, and use it instead of MD.
ML - Mask Decimal, Left Justify Code

The ML code formats and scales numbers, dollars, and amounts, with left justification.

**Syntax**

\[ ML\{n\{m\}{Z}\}{,\}{\$}\{c\}\{(format\ space\ mask) \} \]

- **n** specifies the number of digits (0-9) to print following the decimal point, with rounding; n must be a single digit. If n is omitted, 0 is assumed; if 0 is specified or assumed, no decimal point is printed.
- **m** specifies the scaling factor (0-9); the value is descaled by dividing it by the specified mth power of 10; m is the number of implied digits to the right of the decimal point for the stored value.
- **Z** specifies that leading zeros are to be suppressed.
- **, comma; specifies insertion of a comma between every thousands position of the value.
- **$** appends a dollar sign ($) prior to the value.
- **C** credit indicator code; can be one of the following:
  - **C** causes the letters CR to follow negative values, two blanks to follow positive or zero values.
  - **D** causes the letters DB to follow positive values, two blanks to follow negative or zero values.
  - **E** causes negative values to be enclosed inside angle brackets, as in `<100>`: a blank precedes and follows positive or zero values.
  - **M** causes a minus sign (-) to follow negative values, a blank to follow positive or zero values.
  - **N** causes the minus sign (-) on negative values to be suppressed.
- **(format mask)** controls field length and fill characters: it consists of format codes and literal data. A format code can be one or more of the following, enclosed in parentheses:
**ML - Mask Decimal, Left Justify Code**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#n</td>
<td>left-justifies value in a field of n blanks.</td>
</tr>
<tr>
<td>*n</td>
<td>left-justifies value in a field of n asterisks.</td>
</tr>
<tr>
<td>%n</td>
<td>left-justifies value in a field of n zeros.</td>
</tr>
<tr>
<td>&amp;n</td>
<td>left-justifies value in a field of n blanks.</td>
</tr>
<tr>
<td>x(n)</td>
<td>any other characters, including parentheses and dollar signs, are displayed exactly as specified. Each character adds one to the number of characters displayed in the result. See the examples.</td>
</tr>
</tbody>
</table>

**Description**

The parameters must be specified in the order given.

Except for justifying fields in a format mask, the ML code and MR codes convert an attribute value in the same way. If format masks are specified, the ML codes left-justifies data in the mask. (Justification in the listing and for sorting depend on the V/TYP code in attribute 9 of the attribute definition item.)

The internal format of a masked field is the actual data characters only, with the mask stripped off. For example, if a social security number field has a mask of `ML(%3-%2-%4)` the data is stored as nine numbers (`nnnnnnnnn`) but displayed as eleven characters (`nnn-nn-nnnn`).
<table>
<thead>
<tr>
<th>Code</th>
<th>Attribute Value</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML0</td>
<td>4999</td>
<td>4999</td>
</tr>
<tr>
<td>ML2</td>
<td>4999</td>
<td>49.99</td>
</tr>
<tr>
<td>ML02</td>
<td>4999</td>
<td>50</td>
</tr>
<tr>
<td>ML20,</td>
<td>4999</td>
<td>4,999.00</td>
</tr>
<tr>
<td>ML2E</td>
<td>-123400</td>
<td>&lt;1234.00&gt;</td>
</tr>
<tr>
<td>ML2M</td>
<td>-123400</td>
<td>1234.00-</td>
</tr>
<tr>
<td>ML2,M</td>
<td>-123400</td>
<td>1,234.00-</td>
</tr>
<tr>
<td>ML4</td>
<td>987654</td>
<td>98.7654</td>
</tr>
<tr>
<td>ML2Z</td>
<td>001234</td>
<td>12.34</td>
</tr>
<tr>
<td>ML (%10)</td>
<td>55</td>
<td>5500000000</td>
</tr>
<tr>
<td>ML (*5)</td>
<td>55</td>
<td>55***</td>
</tr>
<tr>
<td>ML, S (#9)</td>
<td>123456</td>
<td>$123,456</td>
</tr>
<tr>
<td>ML, S (#9)</td>
<td>123456</td>
<td>$123,456</td>
</tr>
<tr>
<td>ML, S (*20)</td>
<td>123456789</td>
<td>$123,456,789****</td>
</tr>
<tr>
<td>ML (#2^3#2^#4)</td>
<td>123456789</td>
<td>12^^34^5678</td>
</tr>
<tr>
<td>ML (%%-%%%-%)</td>
<td>123456789</td>
<td>123-45-6789</td>
</tr>
<tr>
<td>ML (#3-#4 EXT.#2)</td>
<td>123456789</td>
<td>123-4567 EXT.89</td>
</tr>
</tbody>
</table>
MP - Mask Packed Decimal Code

The MP code converts ASCII values to their unpacked decimal equivalent values.

Syntax

MP

Description

Data stored in packed format should always be unpacked for output. Unconverted values will not display properly on a terminal.

When an MP code is used, Ultimate RECALL automatically unpacks data for output. If a BASIC program is accessing an attribute that contains packed data, you should unpack the data via an MP code in an OCONV statement, and pack the data via an MP code in an ICONV statement.

For conversion of input data, the MP code directs Ultimate RECALL to convert the input value to its packed decimal format. Only valid decimal digits (0-9) and signs (+ or -) are recognized for conversion purposes. The MP conversion combines each pair of 8-bit ASCII digits into a single packed 8-bit digit by stripping off the 4 high-order bits of each ASCII digit and storing the 4 low-order bits into one nibble of the packed 8-bit digit.

Each digit is packed sequentially from right to left and strung onto the packed value. If an odd number of nibbles results, a high-order 4-bit nibble of '0' is added.

Numbers with leading plus (+) signs or no signs are considered to be positive. Numbers with leading minus (-) signs are considered to be negative. The low-order nibble of the first packed-format digit in the converted value is used to store the sign of the number, that is, the rightmost digit indicates the sign of the number. The sign is stored as one of the following:

- D indicates a negative number.
- F indicates a positive number.

For conversion of attribute data to output format, the MP code directs Ultimate RECALL to unpack each single packed 8-bit byte (containing two 4-bit digits) into a pair of 8-bit ASCII digits. The low order byte, however, is unpacked as one digit and the sign of the number.

Ultimate RECALL User's Guide
Confidential and Proprietary to The Ultimate Corp.
## Processing Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Attribute Value</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP</td>
<td>X'342F'</td>
<td>342</td>
</tr>
<tr>
<td>MP</td>
<td>X'343D'</td>
<td>-343</td>
</tr>
<tr>
<td>MP</td>
<td>X'01100F'</td>
<td>1100</td>
</tr>
<tr>
<td>MP</td>
<td>X'3D'</td>
<td>-3</td>
</tr>
<tr>
<td>MP</td>
<td>X'010D'</td>
<td>-10</td>
</tr>
<tr>
<td>MP</td>
<td>X'1F'</td>
<td>1</td>
</tr>
<tr>
<td>MP</td>
<td>X'723F'</td>
<td>723</td>
</tr>
<tr>
<td>MP</td>
<td>X'099F'</td>
<td>99</td>
</tr>
<tr>
<td>MP</td>
<td>X'78764D'</td>
<td>-78764</td>
</tr>
</tbody>
</table>
MR - Mask Decimal, Right Justify Code

The MR code formats and scales numbers, dollars, and amounts, with right justification.

Syntax

```
MR {n{m}{Z}{,}{$}{c}} (format mask) }
```

- **n**
  - specifies the number of digits (0-9) to print following the decimal point, with rounding; **n** must be single digit. If **n** is omitted, 0 is assumed; if 0 is specified or assumed, no decimal point is printed.

- **m**
  - specifies the scaling factor (0-9); the value is descaled by dividing it by the specified mth power of 10; **m** is the number of implied digits to the right of the decimal point for the stored value.

- **Z**
  - specifies that leading zeros are to be suppressed

- **, comma**
  - specifies insertion of a comma between every thousands position of the value.

- **$**
  - appends a dollar sign ($) prior to the value.

- **c**
  - credit indicator code: can be one of the following:
    - **C** causes the letters CR to follow negative values, two blanks to follow positive or zero values.
    - **D** causes the letters DB to follow positive values, two blanks to follow negative or zero values.
    - **E** causes negative values to be enclosed inside angle brackets, as in <100>; a blank precedes and follows positive or zero values.
    - **M** causes a minus sign (-) to follow negative values, a blank to follow positive or zero values.
    - **N** causes the minus sign (-) on negative values to be suppressed.

-(format mask)-

- controls field length and fill characters: it consists of format codes and literal data. A format code can be one or more of the following, enclosed in parentheses:

Ultimate RECALL User's Guide
Confidential and Proprietary to The Ultimate Corp.
**Processing Codes**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#n</td>
<td>right-justifies value in a field of n blanks.</td>
</tr>
<tr>
<td>*n</td>
<td>right-justifies value in a field of n asterisks.</td>
</tr>
<tr>
<td>%n</td>
<td>right-justifies value in a field of n zeros.</td>
</tr>
<tr>
<td>&amp;n</td>
<td>right-justifies value in a field of n blanks.</td>
</tr>
<tr>
<td>x{n}</td>
<td>any other characters, including parentheses and dollar signs, are displayed exactly as specified. Each character adds one to the number of characters displayed in the result. See the examples.</td>
</tr>
</tbody>
</table>

**Description**

The parameters must be specified in the order given.

Except for justifying fields in a format mask, the ML code and MR codes convert an attribute value in the same way. If format masks are specified, the MR codes right-justifies data in the mask. (Justification in the listing and for sorting depend on the justification code in attribute 9 of the attribute definition item.)

The internal format of a masked field is the actual data characters only, with the mask stripped off. For example, if a social security number field has a mask of MR(%3-%2-%4) the data is stored as nine numbers (nnnnnnnnnn) but displayed as eleven characters (nnn-nn-nnnn).

If a mask used in an MR processing code will produce padding on the left and if a wild card character (that is, | or | or ^) is likely to be specified in the selection criteria, the processing code should not be used as a conversion. If there is padding, a wild card will never match. If you need the MR processing code, specify it as a correlative instead; then each value on file will be converted to match the input value rather than vice versa. Wild cards can be used to match actual characters.
## MR - Mask Decimal, Right Justify Code

<table>
<thead>
<tr>
<th>Code</th>
<th>Attribute Value</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>MR0</td>
<td>4999</td>
<td>4999</td>
</tr>
<tr>
<td>MR2</td>
<td>4999</td>
<td>49.99</td>
</tr>
<tr>
<td>MR02</td>
<td>4999</td>
<td>50</td>
</tr>
<tr>
<td>MR20,</td>
<td>4999</td>
<td>4,999.00</td>
</tr>
<tr>
<td>MR2E</td>
<td>-123400</td>
<td>&lt;1234.00&gt;</td>
</tr>
<tr>
<td>MR2M</td>
<td>-123400</td>
<td>1234.00-</td>
</tr>
<tr>
<td>MR2,M</td>
<td>-123400</td>
<td>1,234.00-</td>
</tr>
<tr>
<td>MR4</td>
<td>987654</td>
<td>98.7654</td>
</tr>
<tr>
<td>MR2Z</td>
<td>001234</td>
<td>12.34</td>
</tr>
<tr>
<td>MR(%10)</td>
<td>55</td>
<td>00000000055</td>
</tr>
<tr>
<td>MR(*5)</td>
<td>55</td>
<td>***55</td>
</tr>
<tr>
<td>MR,S(#9)</td>
<td>123456</td>
<td>$123,456</td>
</tr>
<tr>
<td>MR,(S#9)</td>
<td>123456</td>
<td>$123,456</td>
</tr>
<tr>
<td>MR,S(*20)</td>
<td>123456789</td>
<td>*********$123,456</td>
</tr>
<tr>
<td>MR(#2^3#2^4)</td>
<td>123456789</td>
<td>23^^^45^^^6789</td>
</tr>
<tr>
<td>MR(#3#4#4#%#)</td>
<td>123456789</td>
<td>123-45-6789</td>
</tr>
<tr>
<td>MR(#3#4 EXT.#2)</td>
<td>123456789</td>
<td>123-4567 EXT.89</td>
</tr>
</tbody>
</table>
Processing Codes

MT - Mask Time Code

The MT code converts the time of day from internal to external format. It is generally used as a conversion.

Syntax

\[ \text{MT}\{H\}\{S\} \]

- \(H\) specifies a 12-hour format instead of the 24-hour (military) format. If omitted, the 24-hour format is used.
- \(S\) specifies that seconds will be output in the time of day. If omitted, seconds are not listed.

Description

The internal time format is the number of seconds from midnight. The external time format can be in either military (24-hour) or 12-hour format. For example, if the time is four minutes and one second before midnight, it is represented in military format as 23:55:59 and in 12-hour format as 11:55:59PM.

The MTH or MTHS codes specify a 12-hour external format. For input conversions, AM times must be entered with AM immediately following the numeric time; PM times must be entered with PM immediately following the numeric time. For output conversions, Ultimate RECALL always prints AM or PM immediately following the numeric time.

The MTS or MT codes specify a 24-hour external format, and AM or PM are not valid on input.

For input conversions, illegal input values are converted to null. For output conversions, all illegal values are output as 00:00; a null value is output as a null.

**Note:** In 12-hour format, midnight is 12:00AM and noon is 12:00PM.
### Code Attribute Values

<table>
<thead>
<tr>
<th>Code</th>
<th>Attribute Value</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT</td>
<td>43203</td>
<td>12:00</td>
</tr>
<tr>
<td>MTH</td>
<td>43203</td>
<td>12:00PM</td>
</tr>
<tr>
<td>MTS</td>
<td>43203</td>
<td>12:00:03</td>
</tr>
<tr>
<td>MTHS</td>
<td>43203</td>
<td>12:00:03PM</td>
</tr>
<tr>
<td>MT</td>
<td>(null)</td>
<td>(blank)</td>
</tr>
</tbody>
</table>
MX - Mask Hexadecimal Code

The MX code converts ASCII character strings to their equivalent hexadecimal (base 16) values.

Syntax

MX

Description

The MX code examines the value and converts each byte into two hexadecimal digits. (Hexadecimal digits are in the range 0-9, A-F.)

<table>
<thead>
<tr>
<th>Code</th>
<th>Attribute Value</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>MX</td>
<td>(4 spaces)</td>
<td>20202020</td>
</tr>
<tr>
<td>MX</td>
<td>+1</td>
<td>2B31</td>
</tr>
<tr>
<td>MX</td>
<td>&lt;ESC&gt;A</td>
<td>1B41</td>
</tr>
</tbody>
</table>
P - Pattern Match Code

The P code restricts output to values that match a specified pattern. If the value does not meet the criteria, a null attribute is returned.

Syntax

\[ \text{P(pattern)\{};(\text{pattern})\} \]

(pattern) contains one or more pattern elements to specify a match. The pattern elements are:

- \( nA \) tests for \( n \) alphabetic characters.
- \( nN \) tests for \( n \) numeric characters.
- \( nX \) tests for \( n \) characters of any type.
- literal tests for specified literal string of characters.

Each complete pattern is enclosed in parentheses.

; delimiter separating multiple complete patterns: functions as an OR connective.

Description

The attribute is tested against the specified patterns. The number of characters specified by \( n \) must match exactly the number of characters in the string to be compared. If \( n \) is 0, the value matches the pattern if there are zero or more characters in the string that match the type, regardless of the number of characters in the string.

Two alternative patterns can be specified; the two patterns are separated by a semicolon. The value is tested against both patterns: if it matches either pattern, the test is satisfied.

The result is either the entire value (pattern criteria has been met) or a null value (pattern criteria has not been met). This value is then used to process the statement or fill in the value for the current item's detail line of output.
### Processing Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Attribute Value</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>P (2N)</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>P (3N4A)</td>
<td>123 Main</td>
<td>(null)</td>
</tr>
<tr>
<td>P (3N4A)</td>
<td>123 Main</td>
<td>123 Main</td>
</tr>
<tr>
<td>P (0N4A)</td>
<td>123 Main</td>
<td>123 Main</td>
</tr>
<tr>
<td>P (ULT-3A)</td>
<td>ULT-CAL</td>
<td>ULT-CAL</td>
</tr>
<tr>
<td>P (2X*4X)</td>
<td>67*AB21</td>
<td>67*AB21</td>
</tr>
<tr>
<td>P (2X*4X)</td>
<td>ZZ*1224</td>
<td>ZZ*1234</td>
</tr>
<tr>
<td>P (2X*4X)</td>
<td>00*1234</td>
<td>00*1234</td>
</tr>
<tr>
<td>P (2X*4X)</td>
<td>00*12345</td>
<td>(null)</td>
</tr>
<tr>
<td>P (1A):(2N)</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>P (1A):(2N)</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>P (1A):(2N)</td>
<td>T1</td>
<td>(null)</td>
</tr>
<tr>
<td>P (2N$2A-3X)</td>
<td>12SAB-(*)</td>
<td>12SAB-(*)</td>
</tr>
</tbody>
</table>
R - Range Code

The R code tests an attribute in order to ensure that it is within a specified range of values. If the value does not meet the criteria, a null attribute value is returned.

Syntax

\[ R_{n;m} \{,n;m \} \]

- \( n \) - lower limit.
- \( ; \) - separator; any separator (except system delimiters) can be used; however, a minus sign (-) should not be used since the same symbol also refers to negative numbers.
- \( m \) - upper limit.
- , - delimits ranges.

Description

Multiple ranges can be specified by separating each range with a comma.

The attribute is tested against the range specifications. The result is either the entire value (range criteria has been met) or a null value (range criteria has not been met). This value is then used to process the statement or fill in the value for the current item's detail line of output.

<table>
<thead>
<tr>
<th>Code</th>
<th>Attribute Value</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1:5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>R1:5</td>
<td>927</td>
<td>(null)</td>
</tr>
<tr>
<td>R-10;-1</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>R1:100</td>
<td>100A</td>
<td>(null)</td>
</tr>
<tr>
<td>R1:4,6;9</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>R1:4,6;9</td>
<td>5</td>
<td>(null)</td>
</tr>
</tbody>
</table>
S - Substitution Code

The S code substitutes an attribute value with a different value.

Syntax

\[ S; \text{non-null}; \text{null} \]

- **semicolon**: required separator.
- **non-null**: specifies the value to substitute if this attribute is not null or zero.
- **null**: specifies the value to substitute if this attribute is null or zero.

Description

The substitute value for both the null and non-null entries can be one of the following:

- attribute number of the attribute to use.
- literal string enclosed in quotes (" or ").
- asterisk (*) which causes the attribute to be passed through with no substitution.

The entire attribute is replaced. If the attribute being tested is multi-valued, each value or subvalue, if any, is replaced as specified in the substitution. If the replacement attribute is multivalued, each value mark or subvalue mark, if any, is replaced with a space, and the entire attribute is used as the replacement.
<table>
<thead>
<tr>
<th>Code</th>
<th>Attribute Value</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>S;'AFFIRM';'N/A'</td>
<td>Y</td>
<td>AFFIRM</td>
</tr>
<tr>
<td>S;'AFFIRM';'N/A'</td>
<td>(null)</td>
<td>N/A</td>
</tr>
<tr>
<td>S;'AFFIRM';'N/A'</td>
<td>Y{Y}Y</td>
<td>AFFIRM{AFFIRM}N/A</td>
</tr>
<tr>
<td>S;1;2</td>
<td>001 <em>]</em>]*</td>
<td>**</td>
</tr>
<tr>
<td>(where 3 is current attribute)</td>
<td>002 -]-]-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>003 Y</td>
<td></td>
</tr>
<tr>
<td>S;1;2</td>
<td>001 <em>]</em>]*</td>
<td></td>
</tr>
<tr>
<td>(where 3 is current attribute)</td>
<td>002 -]-]-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>003 (null)</td>
<td></td>
</tr>
<tr>
<td>S:*;'NO BAL'</td>
<td>00</td>
<td>NO BAL</td>
</tr>
<tr>
<td>S:*;'NO BAL'</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>
Processing Codes

T - Text Extraction Code

The T code extracts a specified number of characters from an attribute.

Syntax

\[ T\{m,\}n \]

\( m \)
starting column number (from left to right). If omitted, the starting character depends on the attribute's justification. If the attribute's justification is L, extraction starts at the first character and proceeds left to right. If the attribute's justification is R, extraction starts at the last character and proceeds right to left. (Justification is specified in attribute 9 of the attribute definition item.)

\( \), required separator if \( m \) is specified.

\( n \)
number of characters to extract.

Description

T codes are usually applied to attributes with fixed fields, or to truncate data to prevent folding (line wrap).

The attribute is examined for the starting column, if specified. If it is not found, null is returned.

The specified number of characters are extracted and used to process the statement or fill in the value for the current item's detail line of output.

<table>
<thead>
<tr>
<th>Code</th>
<th>Justification</th>
<th>Attribute Value</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2</td>
<td>L</td>
<td>30#650</td>
<td>30</td>
</tr>
<tr>
<td>T2</td>
<td>R</td>
<td>30#650</td>
<td>50</td>
</tr>
<tr>
<td>T4,2</td>
<td>L</td>
<td>30#650</td>
<td>65</td>
</tr>
<tr>
<td>T4,2</td>
<td>R</td>
<td>30#650</td>
<td>65</td>
</tr>
<tr>
<td>T8,2</td>
<td>L</td>
<td>30#650</td>
<td>(null)</td>
</tr>
</tbody>
</table>
Tfile - Translation File Code

The Tfile code translates an attribute by comparing or replacing it with a value from a specified file, or verifies the existence of the value in the specified file.

Syntax

Tfilename;subcode(vmc);in-amc;out-amc(;;break-amc)

filename specifies the file name to use in the translation process.
;
semicolon; required separator.
subcode subcode to use in the translation process; can be one of the following:

C specifies conversion if possible (the conversion item exists and the attribute is not null). Otherwise, the original value is used.
I input verify only; functions like V for input and like C for output.
O output verify only; functions like C for input and like V for output.
V verifies that the conversion item and attribute exist in the translation file. If they do, the untranslated value is returned. If the item does not exist or if the conversion attribute is null, the following message is returned:

[708] 'item-ID' cannot be converted.
X specifies conversion if possible (the conversion item exists and the attribute is not null). Otherwise, a null value is returned.

vmc used with subcode to specify the number of the value in the translation item to use as the replacement value when the translated attribute is multivalued.
in-amc for input conversions, specifies the number of the attribute in the translation item to use as the replacement value for the attribute; it can be null if no input conversion situations will occur.
out-amc specifies the number of the attribute in the translation item to use as the replacement values for the attribute for all
correlatives and for all output conversions from intermediate format values to output format except when a break-amc is present and the statement specifies this attribute with a BREAK-ON or TOTAL modifier.

break-amc specifies number of the attribute in the translation item to use as the replacement value for a BREAK-ON or TOTAL attribute on break lines only. Valid only if the Ttile code is used as a conversion (attribute 7).

**Description**

A Ttile code is usually applied to attributes where the data is already stored in another file. Use of the code prevents duplication of the data. For example, the Ttile code could reference an attribute in your customer file that contains the customer name. The attribute in the item you are defining could be a number that is the item-ID of the item in the customer file. The Ttile code then translates that number and returns the customer name. That is, the stored value is a reference to the attribute name in the other file.

The translation value replaces the original value, according to the subcode criteria, and the new value is used to process the statement or fill in the value for the current item’s detail line of output.

The V subcode is usually specified as part of the selection criteria in a SELECT statement to determine if the item exists in the translation file. Then another statement using an attribute definition item with the C or X code is specified to list the selected items.

The value of the attribute in the current file is examined and used as the item-ID to match in the specified file (the translation file). Ultimate RECALL then retrieves the values of the in-amc and out-amc. The in-amc is applied only if input conversion is needed to compare input values with attributes with conversion codes. If no in-amc is specified, no input translation takes place.

The out-amc is applied for both correlative and conversion processing, except on break lines when a break-amc is specified. The break-amc is used only on break lines, if any.

The in-amc and out-amc are usually the same attribute number, or else the in-amc is left null.
The break-arcne allows break values or totals to be translated as well as the attribute's individual values. In order for the translated break value to be displayed, use the 'v' option in the text specified with BREAK-ON in the Ultimate RECALL statement.

If the translation attribute is multivalued, and a value count (vmc) is specified, that value is returned. If vmc is not specified, all values are returned with a blank as the separator character.

If a Tfile code is used as part of an F code, the attribute that is translated is taken from the value in stack2. For example, the following code uses the value in attribute 3 to determine the item-ID in the SALES file, moves the value of attribute 1 of the SALES item into stack1, then multiplies that by the value in stack2 (from attribute 1):

F:1:3:(TDict SALES:X:1:1):*

Assume the ZIP file contains the following items:

<table>
<thead>
<tr>
<th>item-ID</th>
<th>92714</th>
<th>92705</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Irvine</td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>CA</td>
<td>CA</td>
</tr>
</tbody>
</table>

When the following data is used, the results for output are as shown (note that 92716 is not in the ZIP file):

<table>
<thead>
<tr>
<th>Code</th>
<th>Attribute Value</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>TZIP:C:1:1</td>
<td>92714</td>
<td>Irvine</td>
</tr>
<tr>
<td>TZIP:C:1:1</td>
<td>92716</td>
<td>92716</td>
</tr>
<tr>
<td>TZIP:I:1:1</td>
<td>92714</td>
<td>Irvine</td>
</tr>
<tr>
<td>TZIP:I:1:1</td>
<td>92716</td>
<td>92716</td>
</tr>
<tr>
<td>TZIP:O:1:1</td>
<td>92714</td>
<td>Irvine</td>
</tr>
<tr>
<td>TZIP:O:1:1</td>
<td>92716 (708) '92716' cannot be converted.</td>
<td></td>
</tr>
<tr>
<td>TZIP:V:1:1</td>
<td>92714</td>
<td>92714</td>
</tr>
<tr>
<td>TZIP:V:1:1</td>
<td>92716 (708) '92716' cannot be converted.</td>
<td></td>
</tr>
<tr>
<td>TZIP:V:1:1</td>
<td>92705 (708) '92705' cannot be converted.</td>
<td></td>
</tr>
<tr>
<td>TZIP:X:1:1</td>
<td>92714</td>
<td>Irvine</td>
</tr>
<tr>
<td>TZIP:X:1:1</td>
<td>92716</td>
<td>(null)</td>
</tr>
</tbody>
</table>
### Processing Codes

```
:CT DICT INVOICE VERIFY.ZIP

VERIFY.ZIP
001 A
002 9
003
004
005
006
007
008 A9!"1","2" (Z\[\(\);1)
009 1
010 20

:SELECT INVOICE WITH VERIFY.ZIP

6 items selected.

:SORT INVOICE COMPANY CITY STATE ZIP

INVOICE... COMPANY... CITY... STATE... ZIP

<table>
<thead>
<tr>
<th>Invoice</th>
<th>Company</th>
<th>City</th>
<th>State</th>
<th>ZIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1681</td>
<td>KelleySoftware</td>
<td>IA</td>
<td>IA</td>
<td>60686</td>
</tr>
<tr>
<td>1682</td>
<td>KelleySoftware</td>
<td>IA</td>
<td>IA</td>
<td>60686</td>
</tr>
<tr>
<td>1683</td>
<td>ServiceSoftware</td>
<td>IA</td>
<td>IA</td>
<td>60660</td>
</tr>
<tr>
<td>1686</td>
<td>QualitySoftware</td>
<td>IA</td>
<td>IA</td>
<td>60660</td>
</tr>
<tr>
<td>1687</td>
<td>ServiceSoftware</td>
<td>IA</td>
<td>IA</td>
<td>60622</td>
</tr>
<tr>
<td>1696</td>
<td>UniversalSoftware</td>
<td>IA</td>
<td>IA</td>
<td>60632</td>
</tr>
</tbody>
</table>

6 items listed.
```
V - Item-ID Extraction Code

The V code specifies an attribute that contains item-IDs to be extracted by the WITHIN modifier. For information on the WITHIN modifier, see Chapter 2.

Syntax

```
V;;amc
;;  two semicolons; required separators.
amc  number of attribute that contains the item-IDs.
```

Description

The V processing code defines the attribute to explode in order to determine the next items to use in the report.

Note that two semicolons are required.

This processing code is valid only as a correlative (attribute 8) in a file definition item. It is not valid in an attribute definition item.
6 Using Ultimate RECALL with Report Forms

You can format reports by using special forms output specifications in the Ultimate RECALL LIST and SORT commands. Multivalued attributes can be output within windows, or subpages. You can print the report on a standard background specified in the statement, or you can format it for preprinted forms.

Figure 6-1 shows an example of a report that can be produced using the Ultimate RECALL forms capability. For information on the statement that created the report, see the section, Example of Form.

Note: In the following descriptions, the word form refers to a report for a single item. If the report for that item exceeds one page, the form is said to be multipaged. A multipaged form results only if a window is specified and there are more values than will fit in the specified number of rows in the window.
Using Ultimate RECALL with Report Forms

**Invoice:** 1682

---

**The XYZ Firm**

12345 Mac Arthur Street  
Los Angeles CA 91777
213/555 1212  
fax 213/555 7878

*Supplying all your food service needs*

<table>
<thead>
<tr>
<th>Company Name</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kelley Brothers</td>
<td>12345 Main Street</td>
<td>Anaheim</td>
<td>CA</td>
<td>92006</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contact</th>
<th>Date</th>
<th>Terms</th>
<th>Ship Via</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jerry</td>
<td>08/24/92</td>
<td>Net 30</td>
<td>Truck</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Product Number</th>
<th>Description</th>
<th>Unit Price</th>
<th>Total Price</th>
<th>Delivery Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>7055</td>
<td>Diet Cola</td>
<td>$9.60</td>
<td>$19.20</td>
<td>05/05/92</td>
</tr>
<tr>
<td>1</td>
<td>6032</td>
<td>Orange Juice</td>
<td>$15.40</td>
<td>$15.40</td>
<td>08/18/92</td>
</tr>
<tr>
<td>1</td>
<td>6065</td>
<td>Cranberry Juice</td>
<td>$12.80</td>
<td>$12.80</td>
<td>08/18/92</td>
</tr>
<tr>
<td>1</td>
<td>6068</td>
<td>Grapefruit Juice</td>
<td>$12.80</td>
<td>$12.80</td>
<td>08/18/92</td>
</tr>
<tr>
<td>1</td>
<td>5011</td>
<td>Hot Chocolate Regular</td>
<td>$7.35</td>
<td>$7.35</td>
<td>08/18/92</td>
</tr>
<tr>
<td>3</td>
<td>5015</td>
<td>Hot Chocolate Sugar Free</td>
<td>$5.60</td>
<td>$16.80</td>
<td>08/18/92</td>
</tr>
</tbody>
</table>

**Total** $84.15

---

*Figure 6-1. Sample Forms Report*

---

6-2  
Ultimate RECALL User's Guide  
Confidential and Proprietary to The Ultimate Corp.
Features of Report Forms

The following is a summary of features that forms output supports:

- each item is printed on a separate page.
- forms can be multipaged, that is, data from a single item can be reported on multiple pages.
- multivalued attributes can be placed within a window (subpage) of a single form. Up to six separate window sets can be defined per statement, with any number of attributes in the window.
- special options are available for forms alignment, placement of background data on the form, output of multiple items on a single page, and page number resetting.
- check numbers can be automatically generated for an audit trail, and the check numbers automatically updated in an audit file.
- an optional background can be defined and printed on the form on which the report data is displayed.
- the width and length of a form are the same as a standard Ultimate RECALL report, that is, the current settings in the TERM command for Page Width and Page Length. The default printer page width is 132 characters (132 columns on a form). The default printer page length is 60 lines (60 rows on a form). (For more information on TERM, see the Ultimate System Commands Guide.)
- on forms output, column headings and item-IDs are not displayed. However, the standard Ultimate RECALL report heading line is used unless a HDR-SUPP or HEADING modifier is present in the statement. There is no footing unless a FOOTING modifier is present in the statement.
- the first line of a form (line 0) is row 0. Line 1 of a form is called row 1, line 2 is row 2, and so on. The first position on a line (character 0) is called column 0, character 1 is column 1, and so on.

Rows 2 through the row specified as printer page length are always available for printing. Row 0 of a form is available only for headings. Row 1 is available for printing if the HDR-SUPP or HEADING modifier is specified. The standard Ultimate RECALL heading uses rows 0 and 1; therefore, if neither HDR-SUPP nor HEADING is specified, row 1 is not available for printing.
Using Ultimate RECALL with Report Forms

- characters 0 to the maximum number of characters are available for printing.
- there is no end-of-report message; that is, the message 'n' items listed, is not displayed.

The forms capability has the following restrictions:
- controlling and dependent attribute sets may cause incorrect reports. If a controlling attribute has no (or null) value, dependent attribute values are not retrieved and they are ignored on the report. This changes the sequence of attribute data passed to the forms program, and can result in incorrect placement of data on the form.
- attributes with Tfile (file translation) codes used as correlatives (attribute 8 of the attribute definition item) may cause incorrect reports. If the correlative returns a null value, the attribute is skipped entirely. One too few attributes will be passed to the forms program, which results in incorrect placement of data on the form.

For more information on these two restrictions, see the section, Null Values.
The forms capability is available as part of the LIST or SORT statement. A forms report is produced by specifying the output specifications in the statement as *forms expressions*.

The syntax of a forms expression varies, depending on the type of information to be displayed:

- literal text and single-valued attributes use the following syntax.
- multi-valued attributes should be specified in windows. For more information, see the section Windows.

### Syntax

```plaintext
@{p}(c,r{"message"}):name{"1,2}\n@{p}(c,r):"text"
```

- `@` signals Ultimate RECALL that a forms expression follows. The first format positions the data in *attrib-name* on the form. The second format positions *text* on the form.
- `p` print specification code. The following codes are available:
  - `A` prints attribute on all pages; this is the default.
  - `C` prints automatic incrementing check number.
  - `F` prints attribute on first page only.
  - `L` prints attribute on last page only.
  - `M` prints continuation message on all but last page of each form. Message must be a string enclosed in single or double quotes, and can contain blanks.
  - For more information on these codes, see the section, Print Specifications.
- `c` column number, where first column is 0.
- `r` row (line) number, cannot be 0 (which is reserved for the heading).
- `message` literal string to print before the attribute; used with the M print specification code.
- `:` colon: required separator.
Using Ultimate RECALL with Report Forms

name
attribute name; that is, the item-ID of the attribute definition item in the file dictionary.

[1,z]
substring specification. It must begin with the number 1; z is the number of characters to extract from the attribute value.

text
text to display in specified position; must be enclosed in single (') or double (") quotes, and can contain blanks.

Note: Except for a delimited text entry, blanks are not allowed within an expression. Blanks are used as separators between expressions.

Description
A forms expression directs Ultimate RECALL to place an attribute or string of output at a specified row and column on the report page. The attribute definition items used for forms expressions are exactly the same as for any other Ultimate RECALL command.

Because a forms expression specifies the location where the data is to be printed, the output specifications for forms-type reports can be named in any order. One or more literal strings can be placed on the form.

Item-IDs are not automatically placed on the form. If item-IDs are needed on the form, an attribute definition item can be defined for attribute 0 (which is the item-ID); use that attribute name in the forms expression that positions the item-ID. For example, the following is an attribute definition for the item-ID in the INVOICE file:

```
item-ID: INVOICE#
  001 A
  002 :
  003 Invoice Number
  004
  005
  006
  007
  008
  009 L
  010 10
```

The item-ID could be placed on a form using a forms expression similar to the following:

```
@ (3,5):INVOICE#
```
<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@ (15, 7) : NAME</td>
<td>prints the value of NAME starting at column 15 on row 7 of every page of the form set.</td>
</tr>
<tr>
<td>@A (5, 15) : DESC</td>
<td>prints the value of DESC starting at column 5 on row 15 of every page of the form set.</td>
</tr>
<tr>
<td>@M (1, 12, &quot;** MORE **&quot;) : N</td>
<td>prints the message &quot;** MORE **&quot; on all but the last page of each form. Prints the value of N on last page.</td>
</tr>
</tbody>
</table>
Using Ultimate RECALL with Report Forms

Print Specifications

Ultimate RECALL uses the print specification code and location \((c,r)\) to determine how and where to print the attribute data or text string on the form.

A (All Pages) Print Specification Code

The A print specification code specifies that the associated data is to be printed on every page of a multipaged form. The code has the following format:

\[ @A(c, r): attrib-name \]

The A print specification code is the default code; if no print code is specified, A is assumed.

The A print specification code is useful for attributes such as ship-to name and address, which should print on each page.

C (Check/Invoice Number) Print Specification Code

The C print specification code generates an automatically incrementing number for each form and creates an audit file item as each form is generated. The data in the specified attribute is stored in the audit file item and the current generated number is printed at the location specified. No attribute or text data is printed on the form other than the current sequence number. The audit file must exist.

Any number of C codes can be specified.

If the C code is specified, the following prompt is displayed before output starts:

Audit File:

Enter the filename where the items for the audit trail are to be stored. The following prompt is displayed:

Starting Number:

Enter the number to print on the first form. This number is typically a check or invoice number. If forms alignment is specified, this number is incremented for each forms alignment page.
After the prompts have been answered, if forms alignment has been specified, the alignment output is generated, and the operator can align the forms as needed.

For more information on the audit file, see the next subsection.

**F (First Page) Print Specification Codes**

**L (Last Page) Print Specification Codes**

The **F** print specification code specifies that the associated data is printed on only the first (F) page of a multipaged form. The **L** print specification code specifies that the associated data is printed on only the last (L) page of a multipaged form. On all other pages of a multipaged form, nothing is printed for the specified attribute. The codes have the following format:

`@F(c, r):attrib-name`

`@L(c, r):attrib-name`

On forms that are not multipaged, the codes are identical to the **A** code.

The **L** print specification code can be used, for example, to print totals that are to be generated only on the last page of a multipaged form.

**M (Message) Print Specification Code**

The **M** print specification code specifies that message text is printed on every page except the last page of a multipaged form. The data in the specified attribute is printed on only the last page of the form. The code has the following format:

`@M(c, r, "message"): attrib-name`

On forms that are not multipaged, only the data in the specified attribute is printed; the message is not printed.

The **M** code can be used, for example, when printing checks or for informational messages. On checks, where the presence of a multivalued field in the stub can cause a multiple-paged check, the **M** code can be used to print a voiding message on the unused checks. For example, the following code prints the message on all unused checks, at column 20 on line 16.

`@M(20, 16, "* * * VOID  * * VOID  * * VOID  * *")`
Audit File

The filename specified as the audit file must be an existing file. The system creates one item for each number generated during the forms output run. The item-IDs for the audit file items are the numbers generated by the C code. If there is already an item with that item-ID, the item is overwritten and no warning is given.

Ultimate RECALL adds one attribute to the audit item for each C code output value. Thus, if the statement has one C code, there will be one attribute added in each item. If three C codes are used, there will be three attributes added in each audit file item. The format of each item is as follows:

- **item-ID**: sequential number, starting with specified number.
- **001**: code; can be one of the following:
  - A: number was generated by the forms alignment process.
  - 0: number is not on the last page of the form.
  - 1: this number is on the last page of the form.
- **002**: date item was created, in internal format.
- **003**: time item was created, in internal format.
- **004**: data in attribute specified with first C code.
- **005**: data in attribute specified with second C code.
- **006**: data in attribute specified with third C code.
- **007**: ...

On single-page forms, the value of the code in attribute 1 is 1 for all forms except those generated during the forms alignment. On multipaged forms, the code in attribute 1 is 0 for all pages but the last; on the last page it is 1.

Attribute 1 can be useful for control purposes, such as to determine which checks of a multipaged check were voided, which were valid checks, and which were discarded as part of the forms alignment.

The data associated with the attribute specified in the `forms` expression is stored in the audit file in order to identify the individual forms and provide an audit trail. For example, the name on each form could be stored as an identifying attribute.
Assume the following is the attribute definition item for CHECK#:

```
CHECK#
001 A
002 0
003
004
005
006
007
008 A:15:"*":16
009 L
010 5
```

The following print specification code will create an audit trail of check numbers and the values associated with them:

```
3C(50,10):CHECK#
```

When the code is executed in an Ultimate RECALL SORT or LIST statement, the following occurs (operator responses are in bold):

1. Ultimate RECALL prompts for audit file and starting number:
   
   ```
   Audit File>CAUDIT
   Starting number>10042
   ```

2. The check number 10042 is printed on the first form at column 50 on row 10; 10043 is printed on the second check form, and so on.

3. An item is created in CAUDIT for each check number. The item-ID of the first audit item is 10042; the second item-ID is 10043, and so on. In each item, attribute 1 contains a 1. Attributes 2 and 3 contain the system date and time. Attribute 4 (if this is the first 'C' code) contains the data from attributes 15 and 16 of the processed items in the file specified in the Ultimate RECALL statement.
Windows

A window is a portion of a forms page used for displaying multivalued sets of attributes. Several fields of information can be spaced at different column positions across the page within the window. The same type of information can then be repeated on different rows within the window, such as line items in an invoice or individual items in an order.

Syntax

WINDOW @(1,u,v{s}) @(c1{,y1}):attrib-name1{[1,z1]}
...(c{,yn}):attrib-namen{[1,zn]}...

END-WINDOW

WINDOW specifies the beginning of a window and its size. All forms expressions between a WINDOW connective and the next END-WINDOW are considered to be part of the same window. Windows cannot be nested.

1 the number 1; not used by Ultimate RECALL, but included for compatibility with Ultimate UPDATE.

u upper row limit of window. Specifies the number of the first row to be printed within the window.

v lower row limit of window. Specifies the number of the last row to be printed within the window.

s if specified, must be the number 2; a double-depth window is created, so that each window line uses two rows on the form. This allows additional space for printing multivalued fields, or allows double-spacing between lines in a window. If omitted, each window line is printed in one row of the form.

cn starting column for attribute n; no row is specified because the WINDOW parameters set up the number and location of the rows that make up the window.

yn if specified, must be the number 2; positions data in attribute n in the second row of the window. Should be used only if the window is double-depth, that is, if s is also specified.

: colon: required separator.
Windows

attrib-namen  attribute name of the nth attribute to be displayed.

[1,zn]  substring specification. It must begin with the number 1; z is the number of characters to extract from attribute n.

END-WINDOW  specifies the end of the window.

Description

A window expression reserves a group of rows as a window within a page so that multiple values can be listed on separate rows, and the item can be continued to two or more forms pages, if needed. The reserved group of rows is referred to as a subpage set.

A window is considered to include all available columns on the form, although some may contain no data.

When windows are specified, the data is placed across the form at the specified column (c), within the specified rows (u,v) of the window. Multiple values are printed in columns in the window, one below another. When any column in a window is filled and there are more values to output, a new forms page is started.

Up to six independent WINDOW sets can be specified in one Ultimate RECALL forms statement. If any WINDOW specifications cause windows to overlap on the form, no error message is generated. If the rows overlap, but the columns do not, the report is formatted as expected. However, if the columns overlap, attributes may be overwritten.

Multivalued attributes should be placed in a window. If a multivalued attribute is not placed in a window, only the first value is printed on the form.

If windows are used and if the number of multivalues in an attribute exceeds the limit of the window, a new page is printed. If no windows are present in an Ultimate RECALL forms output statement, all the output data from an item is placed on one page.

Only A or null print specification codes are allowed within a window.
The following defines a single-depth window with two fields:

```
WINDOW @(1,4,8) @(5):PROD.NO @(15):DESC @(32):EXT.PRICE END-WINDOW
```

Data is positioned in this window as follows:

<table>
<thead>
<tr>
<th>Column</th>
<th>012345678901234567890</th>
<th>012345678901234567890</th>
<th>012345678901234567890</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>35</td>
<td>36</td>
</tr>
</tbody>
</table>

Data: Diet Cola $19.20

Root Beer $9.90

Apple Juice $30.80

The following defines a double-depth window with three fields. The first two fields are in the first row and the third is in the second row.

```
WINDOW @(1,4,8,2) @(5):PROD.NO @(15):DESC @(32,2):EXT.PRICE END-WINDOW
```

Data is positioned in this window as follows:

<table>
<thead>
<tr>
<th>Column</th>
<th>012345678901234567890</th>
<th>012345678901234567890</th>
<th>012345678901234567890</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>35</td>
<td>36</td>
</tr>
</tbody>
</table>

Data: Diet Cola $19.20

Regular Cola $9.95

Diet Root Beer $9.95

Grape Root Beer $9.90

Apple Juice $30.80
Modifiers in Forms Output

The following Ultimate RECALL modifiers function somewhat differently in forms than in standard Ultimate RECALL reports:

- **BREAK-ON**
- **COL-HDR-SUPP**
- **DBL-SPC**
- **FOOTING**
- **GRAND-TOTAL**
- **HDR-SUPP**
- **ID-SUPP**
- **HEADING**
- **ONLY**
- **SUPP**
- **TOTAL**

**BREAK-ON**

Control breaks in forms output are processed the same as in a standard Ultimate RECALL report, except that a form is printed for each break. Break lines contain only the default break field (*** or specified text) which is displayed where the BREAK-ON attribute data was printed.

The BREAK-ON modifier in a forms report has the following syntax:

```
BREAK-ON @(c,r):attrib-name {"text...{'options'}...text"}
```

If a C (audit file) print specification code is specified, an audit file item is created for every break. If totals are also being generated, the totals will be placed in the corresponding audit file item for each break.

**COL-HDR-SUPP**

The column headings are automatically suppressed on forms, so the COL-HDR-SUPP modifier is not needed.

**DBL-SPC**

The DBL-SPC modifier has no effect.

**GRAND-TOTAL**

The GRAND-TOTAL modifier has no effect.
Using Ultimate RECALL with Report Forms

**HDR-SUPP**
By default, the standard report heading row with page number and report time and date is output on row 1 of the forms report. To suppress this row, use the HDR-SUPP modifier. This makes row 1 of the form available for printing.

If HDR-SUPP is used and no HEADING modifier is present, a blank one-line heading is still reserved on the form. This means that row 0 is never available for data; it is only available for heading information.

**HEADING and FOOTING**
A HEADING or FOOTING modifier causes Ultimate RECALL to reserve the specified rows on the form. These rows reduce the number of available rows for printing data.

If the form statement specifies the 'Z' option (reset page number to 1 for each new multipaged form), you should put the 'P' (page) code in the FOOTING, not the HEADING. This is because headings are printed before the start of the new form and the page number has not yet been reset. Footings, however, are printed after the page number has been reset.

**ID-SUPP**
The item-ID column is automatically suppressed on forms, so the ID-SUPP modifier is not needed.

**ONLY**
The ONLY modifier has no effect.

**TOTAL**
Totals in forms output are processed the same as in a standard Ultimate RECALL report, except that a form is printed for the total page. Each total line is positioned on the form where the TOTALed attribute data was specified.

The TOTAL modifier has the following form:

```
TOTAL @(c,r):attrib-name
```

If a C print specification code is specified for an attribute that is being totaled, the highest numerical item-ID in the file will contain the grand-total of that attribute. The data and totals will be placed in the file and no totals or data will be displayed on the form, unless you specify another forms element.
Options in Forms Output

The following options are available for LIST and SORT commands when forms output is specified:

A  alignment. Allows forms alignment, before printing begins. Meaningful to printer only when the I option of the SP-ASSIGN command is in effect.

B  background. Prints background data before printing the report form data.

M  multiple items per page. Divides the report page into a specified number of subpages.

Z  page number reset for multipaged forms. Resets the page number to 1 for each new form.

The C option, if specified, has no effect.

A Option - Aligning a Form

Forms alignment is a real-time process performed under the operator's control at the time the forms are printed. It is meaningful for output to a printer only when the current SP-ASSIGN command includes the I (immediate print) option.

When the A option is specified, it causes a sample form to be printed, using fields of x's and 9's in place of the actual output data. This sample format is printed on the very first form at the printer or terminal.

After the sample form is printed, the following prompt is displayed:

   Align? (Y=CR/N)

If the fields are properly aligned on the form, begin the regular forms output printing by entering the following:

   N

If the fields are not properly aligned, adjust the printer and repeat the alignment by pressing RETURN (or entering any key except N).
B Option - Defining a Background

The B option is used to print background data on every form that is generated. The background data is stored in an item. Ultimate RECALL prints the background data on the row first, then processes the current item from the file. The background can be overwritten with data from the current item.

When the B option is specified, Ultimate RECALL prompts for the filename:

```
Background file and item>
```

Enter the file and item name.

The data should be stored exactly as it is to appear on the form. Each attribute of the background item is printed on the corresponding row of the form; that is, attribute 001 data is printed on row 1 of the form, attribute 002 on row 2, and so on. Blank rows can be left null.

The background data can contain special printer control characters such as expanded print or underscore features. Note that using control characters can alter the positioning of data on that row of the form; some experimentation may be required to correct for this.

The following item can be used to print the logo on rows 2-4, the word INVENTORY on row 6, the word LISTING on row 7, and the double line on row 8. Row 8 is the last row of the item. The data from the items can then be printed as specified in the forms expressions.

```
001:
002: ...........................................................
003: ...........................................................
004: ...........................................................
005: ...........................................................
006: INVENTORY:
007: LISTING
008: LIST
```

Ultimate RECALL User's Guide
Confidential and Proprietary to The Ultimate Corp.
M Option -
Multiple
Reports per
Page

The M option is used to produce a report in forms format, but with more
than one item on an output page. When the M option is specified,
Ultimate RECALL prompts for the number of rows per item:

Subpage size>

Specify the number of report rows needed for each item, including a
blank row between items. For example, if you are printing five rows in
the form, specify a subpage size of at least 6. Ultimate RECALL
determines the number of items that will fit on each output page based
on the number of rows and print. The number of available print rows is
the value of the page length defined in the current TERM parameter less
the number of heading and footing rows (if any).

Only whole numbers are considered in the result. If an entire item
cannot fit on a page, a new page is started to print that item. For
example, a page length of 60 rows with a heading of three rows and a
footing of two rows results in 55 print rows per page. With a six-row
subpage, each output page can contain up to nine items with five lines of
data and one blank line. With an eleven-row subpage, up to five items
will fit on a page.

Ultimate RECALL does not test the coordinates used in the print
specification codes to ensure that they do not exceed the subpage size.
If any data does not fit within the subpage, it is not displayed.

In the following example, each item uses four rows, so that up to 11
items can print on a single page (including blank lines between items).
Z Option - Resetting Page Number

The Z option resets the page number to 1 at the beginning of each form. With multipaged forms, it may be desirable to do this to allow printing of page numbers relative to each individual multipaged form.

When this option is used, the page number 'P' code should be placed in the FOOTING, not the HEADING. This is because headings are printed before the start of the new form and the page number has not yet been reset. Therefore, the first page of each new form would always contain a page number one higher than the last page of the previous form. Footings, however, are printed after the page number has been reset.
Placing Data

Because each forms expression specifies the location where the data is to be printed, the output specifications for forms-type reports can be named in any order.

However, the following elements of an attribute definition item do have an effect on the placement of data:

- S/AMC, attribute 4.
- V/TYP, attribute 9.
- V/MAX, attribute 10.
- null values in controlling or dependent attributes, or in attributes that are processed with Tfile translation.

S/AMC

S/AMC defines controlling and dependent attributes. In a report, the controlling attribute is always listed first and the dependent attributes are listed in the order they are specified in the S/AMC attribute for the controlling attribute, regardless of the order in which they are specified and the locations in the forms expression.

V/TYP

The (c,r) location specifies the starting character position on the form. If the V/TYP code is L, T, or U, the data is left-justified starting at column c. If the V/TYP code is R, the data is right-justified with the rightmost column at column (c + V/MAX).

If the length of the actual data is greater than V/MAX, the V/TYP determines the display as follows:

- L the data is truncated at number of characters specified in V/MAX.
  For example, if the forms expression is @(3,2):MAKE and the V/MAX for the A-item called MAKE is 10, the last column for MAKE data is (12,2) and no more characters are printed.

- R the data overflows to the left, up to the left margin. For example, if the forms expression is @(3,2):HP and the V/MAX for the A-item called HP is 6, the rightmost position is column 9 of row 2 of the form; the leftmost character depends on the number of characters in the data and the number of columns on the form.
Using Ultimate RECALL with Report Forms

V/MAX

T if the forms expression is not within a window, the data is truncated at V/MAX. If the forms expression is within a window, the data wraps around to the leftmost column of the next row at a word boundary (the last word that would not fit on the current row).

U the data overflows to the right.

Table 6-1 is a summary of the effects of V/TYP and V/MAX on forms.

V/MAX

If V/MAX is greater than zero, the length is checked and the data may fit, fold, or be truncated as determined by the V/TYP. If V/MAX is 0, no data is copied to the form on detail lines. If an attribute with a V/MAX of 0 is used with BREAK-ON, it is displayed as it would have been with any other V/MAX.

Table 6-1 is a summary of the effects of V/TYP and V/MAX on forms.
### Table 6-1. Effects of V/TYP and V/MAX on Forms

<table>
<thead>
<tr>
<th>Condition</th>
<th>V/TYP</th>
<th>On Standard Reports</th>
<th>On Forms</th>
</tr>
</thead>
<tbody>
<tr>
<td>data &lt;= V/MAX</td>
<td>L, T, U</td>
<td>prints on one row, left-justified.</td>
<td>prints on one row, left-justified.</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>prints on one row, right-justified.</td>
<td>prints on one row, right-justified.</td>
</tr>
<tr>
<td>data &gt; V/MAX</td>
<td>L</td>
<td>folds to new row at V/MAX character.</td>
<td>truncates at V/MAX character.</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>folds to new row at word boundary prior to V/MAX.</td>
<td>if in window, folds to new row at word boundary; if not in window, truncates at V/MAX character.</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>overflows to right.</td>
<td>overflows to right.</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>overflows to left.</td>
<td>overflows to left.</td>
</tr>
<tr>
<td>V/MAX = 0</td>
<td>all</td>
<td>if S/NAME is \ (no tag), data is suppressed on detail lines; otherwise data is displayed using length of S/NAME or, if not specified, length of item-ID to determine number of characters to display.</td>
<td>no length test; all data is suppressed on detail lines, but printed on break rows, if specified.</td>
</tr>
</tbody>
</table>
Null Values

Null values can sometimes cause data to be misplaced if they are in controlling or dependent attributes, or in attributes with Tfile translation. The following describes some considerations if you must use controlling and dependent attributes, or attributes that are processed with Tfile translations.

Controlling or Dependent Attributes

Controlling and dependent attribute sets may require special handling when the controlling attribute is specified within a window. If a controlling attribute is null, no dependent attributes are retrieved. This can change the sequence of data passed to the forms program, and results in incorrect placement of data on the form.

The forms program has no control over the handling of null values in controlling and dependent attributes. The only reason to use these specifications is if a print-limiter is to be used on the controlling attribute, which will automatically limit the dependent attribute data to those sets that meet the limit test.

To take preventive action, you can do one of the following:

• if the controlling and dependent specifications are not needed, they can be removed from the attribute definition items referenced by the Ultimate RECALL command. Alternatively, attribute definition items can be defined specifically for the forms application that do not use controlling and dependent definitions.

• if the controlling and dependent specifications are relevant (print-limiting on the set is needed), that set of attributes must be entered as the last forms expressions in the statement. A statement can have only one controlling/dependent set.

Tfile Attributes

Without special handling, an attribute with a Tfile (file translation) processing code used as a correlative can cause incorrect forms output. If the correlative returns a null value, the attribute is skipped entirely, which means that one too few attributes are passed to the forms program. This results in incorrect placement of data on the form. (A null value could be returned, for example, when an X code is used in the Tfile code.)
The forms program has no control over the handling of null values in this case. To take preventive action, you can do one of the following:

- move the Tfile code from the correlative field (line 8) to the conversion field (line 7) of the attribute definition item.
- change the Tfile code element from X to C or V.
- change the Tfile code to an element within an A or F correlative and append one blank onto the result. This will always return an intermediate value of at least one blank (not null). For example, this following Tfile correlative and A correlative are equivalent (assuming attribute 3 is used as the source data), except the A correlative never returns a null:

```
TINV;X;;10
A:3(TINV;X;;10):" "
```
Using Ultimate RECALL with Report Forms

Example of Form

The example below illustrates the file dictionary items and Ultimate RECALL statement used to produce the report shown in Figure 6-1.

The INVOICE file is used to produce the forms output. Figure 6-2 lists the DICT INVOICE file attribute definition items that are used to format the data on the form. Figure 6-3 shows the PROC that produces the forms statement. (For more information on creating and using PROCs, see the Ultimate PROC Reference Guide.)

<table>
<thead>
<tr>
<th>INVOICE</th>
<th>CDAMC</th>
<th>S/NAME</th>
<th>CONVERSION</th>
<th>CORR=DEFAULT</th>
<th>TP MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDRESS</td>
<td>A 8</td>
<td>Address</td>
<td>L 20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CITY</td>
<td>A 9</td>
<td>City</td>
<td>L 20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPANY</td>
<td>A 1</td>
<td>Company Name</td>
<td>VCT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATE</td>
<td>A 2</td>
<td>Invoice Date</td>
<td>Z20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEL DATE</td>
<td>A 5</td>
<td>Delivery Date</td>
<td>Z20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESC.1</td>
<td>A 991</td>
<td>Description</td>
<td>VCT X 4,3</td>
<td></td>
<td>L 25</td>
</tr>
<tr>
<td>EXT. PRICE</td>
<td>A 91</td>
<td>Ext Price</td>
<td>V203S</td>
<td></td>
<td>L 10</td>
</tr>
<tr>
<td>INVOICE</td>
<td>A 5</td>
<td>Invoice Number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td>A 7</td>
<td>Contact</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRICE.1</td>
<td>A 3</td>
<td>Price</td>
<td>V203S X 7</td>
<td></td>
<td>L 10</td>
</tr>
<tr>
<td>PROD.NO</td>
<td>A 3</td>
<td>Product Number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QTY.1</td>
<td>A 4</td>
<td>Qty</td>
<td>R 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STATE</td>
<td>A 9</td>
<td>State</td>
<td>L 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TERM</td>
<td>A 10</td>
<td>Terms</td>
<td>L 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOT. PRICE</td>
<td>A 90</td>
<td>Tot Price</td>
<td>V203S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VIA</td>
<td>A 11</td>
<td>Ship Via</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZIP</td>
<td>A 9</td>
<td>Zip</td>
<td>R 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6-2. DICT INVOICE Items
Example of Form

```
PRINT.INVOICE
001 PQ
002 * Print out invoices
003 HSORT: INVOICE
004 H $ (13,1): INVOICE#
005 H $ (2,8): COMPANY
006 H $ (24,8): ADDRESS
007 H $ (46,8): CITY
008 H $ (68,8): STATE
009 H $ (72,8): ZIP.
010 H $ (2,12): NAME
011 H $ (24,12): DATE
012 H $ (46,12): TERM
013 H $ (68,12): VIA
014 H WINDOW $ (1,21, 40,2)
015 H $ (2): CITY
016 H $ (9): PROD.
017 H $ (22): DESC.
018 H $ (41): PRICE
019 H $ (58): EXT.PRICE
020 H $ (70): 6
021 H END-WINDOW
022 H $ (2,42, "Continued . . "): MESSAGE
023 H TOTAL $ (58,47): EXT.PRICE
024 H HDR-SUPP
025 IF A7 "?"
026 H (? A
027 P
```

**Figure 6-3. PROC to Create Reports**

The address, city, state, and zip code data are printed only on the first page of a multipaged form. The header is suppressed, allowing the invoice number to be printed at the top of the form.

A 20-line double-spaced window is defined. Ten items can fit on one invoice. If more than ten items are included in an invoice, a second page is started. When more than one page is required, the message Continued . . is printed on all but the last page of the form. The attribute MESSAGE is null and is not printed. The total value of the invoice is printed at the bottom of the last page.

The forms alignment (A) option allows the operator to preview a sample form filled out with XXXXX's to see if the alignment is correct before printing.
Figure 6-4 shows a sample 2-page form. The report was generated by the statement shown in Figure 6-3. This is similar to the report in Figure 6-1, except Figure 6-4 does not show the pre-printed form.

Figure 6-4. Sample Multipaged Form (1 of 2)
<table>
<thead>
<tr>
<th>Quantity</th>
<th>Code</th>
<th>Item</th>
<th>4/6/92</th>
<th>5/6/92</th>
<th>Order Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>6032</td>
<td>Orange Juice</td>
<td>$15.40</td>
<td>$61.60</td>
<td>08/18/92</td>
</tr>
<tr>
<td>1</td>
<td>6068</td>
<td>Grapefruit Juice</td>
<td>$22.80</td>
<td>$22.80</td>
<td>08/18/92</td>
</tr>
<tr>
<td>1</td>
<td>0005</td>
<td>Herb Tea</td>
<td>$20.25</td>
<td>$20.25</td>
<td>08/18/92</td>
</tr>
<tr>
<td>1</td>
<td>2025</td>
<td>Regular Tea</td>
<td>$3.50</td>
<td>$3.50</td>
<td>08/18/92</td>
</tr>
<tr>
<td>6</td>
<td>8036</td>
<td>Cookies</td>
<td>$56.70</td>
<td>$40.20</td>
<td>08/18/92</td>
</tr>
</tbody>
</table>

Figure 6-4. Sample Multipaged Form (2 of 2)
Using Ultimate RECALL with Report Forms

Sequence of Prompts

The forms program can issue a number of prompts, depending on options used in the Ultimate RECALL statement. The following options generate prompts:

- A  alignment option.
- B  background file option.
- M  multiple reports per page option.

The following print specification code generates a prompt:

- C  audit file print specification code.

The prompts that are issued are displayed in the sequence shown in Table 6-2. (This information can be useful if the responses to these prompts are to be stacked by a PROC or a BASIC program.)

Note: If the forms statement is in a PROC or BASIC program and responses to the prompts have been stacked, the prompts are not displayed.

If a slave printer is being used, it is automatically turned off while the prompts and responses are being typed. This prevents the forms from being misaligned.

Table 6-2. Sequence of Prompts

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Condition</th>
<th>Prompt</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B option</td>
<td>Background File &amp; Item&gt;</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>M option</td>
<td>Subpage size&gt;</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>C code</td>
<td>Audit File&gt; Starting number&gt;</td>
<td>Starting number can be stacked, but is normally entered by the operator.</td>
</tr>
<tr>
<td>4</td>
<td>A option</td>
<td>Align? Y=CR/N&gt;</td>
<td>Response cannot be stacked</td>
</tr>
</tbody>
</table>
Ultimate RECALL contains many features that provide great flexibility. This chapter contains some hints on combining these features so you can get precisely the report you want.
Hints

Performance

With large databases, the time required to process a report can sometimes take several minutes. The following suggestions can help improve this time.

File Allocation

Check that your file allocation is correct. Use ISTAT to find out how the items in your files are currently placed in groups. If possible, each group should use no more than one frame. Every time a group uses more than one frame, the system may have to access the disk several times in order to find an item in that group.

If your data items tend to be smaller than the frame size of your system, you should be able to minimize the number of disk accesses by trying to keep the size of groups to one frame. Use HASH-TEST to check for optimum modulos. Then reallocate your files. (File reallocation is described in The Ultimate System Management Guide.)

Specifying Item-IDs

If you specify a relational operator with an item-ID, all item-IDs are accessed and compared, whereas if you specify just the item-ID, only that item is accessed. The latter method is much faster when working with large files and should be used whenever possible. For example, the first statement looks at all items in the INVOICE file, whereas the second statement looks at only the specified item. The same item is retrieved in both cases.

LIST INVOICE = "1682"
LIST INVOICE '1682'

Processing Codes and Sorting

Correlatives are processed before items are sorted. Some processing codes usually do not affect the results of sorting. In order to save processing time, these codes should not be used as correlatives in attribute definition items that will be used in sort keys.

MR ML MC

Other codes may or may not affect the order of sorting. You should analyze attributes that will be used for sorting to see if the processing codes can be used as conversions rather than correlatives.
Using Indexes

Use indexes to create presorted lists. The time to create the index is equivalent to accessing the file in a sorted order without an index. However, once the index is created, the items-IDs are stored in sorted order. Thereafter, accessing the file in sorted order uses the index and the performance is greatly improved.

To create indexes, use the CREATE-INDEX system command, which is described in *The Ultimate System Commands Guide*.
Default Reports

When setting up data files and their dictionaries, it is a good idea to create consecutively numbered attribute definition items for all attributes you wish to see on a standard Ultimate RECALL report. Assign the numbers in the order you wish to display the attributes on the report. This eliminates the need for specifying these attributes every time you want to see the report.
In most applications, since dependent attributes are part of an associated set, they are normally printed only in conjunction with the controlling attribute. For some reports, however, you may wish to output the dependent attribute but not the controlling attribute. For such cases, setting up an alternative attribute definition item may be useful.

For example, in the INVOICE file, the delivery date is dependent on the product number. If the product number is not displayed, neither is the delivery date. The dependent attribute is useful, because if you limit the controlling attribute to certain values, the dependent values are also limited. However, this dependency also means that you cannot print just delivery dates. To solve this, you can set up two attribute definition items for delivery date—one that has the dependency and one that does not.

<table>
<thead>
<tr>
<th>6</th>
<th>non-dependent attribute definition item.</th>
</tr>
</thead>
<tbody>
<tr>
<td>001 A</td>
<td></td>
</tr>
<tr>
<td>002 6</td>
<td></td>
</tr>
<tr>
<td>003 Delivery Date</td>
<td></td>
</tr>
<tr>
<td>004</td>
<td></td>
</tr>
<tr>
<td>005</td>
<td></td>
</tr>
<tr>
<td>006</td>
<td></td>
</tr>
<tr>
<td>007 DEL.DATE</td>
<td></td>
</tr>
<tr>
<td>008</td>
<td></td>
</tr>
<tr>
<td>009 6</td>
<td></td>
</tr>
<tr>
<td>010 9</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8</th>
<th>dependent attribute definition item.</th>
</tr>
</thead>
<tbody>
<tr>
<td>001 A</td>
<td></td>
</tr>
<tr>
<td>002 6</td>
<td></td>
</tr>
<tr>
<td>003 Delivery Date</td>
<td></td>
</tr>
<tr>
<td>004 DEL.DATE</td>
<td></td>
</tr>
<tr>
<td>005</td>
<td></td>
</tr>
<tr>
<td>006</td>
<td></td>
</tr>
<tr>
<td>007 6</td>
<td></td>
</tr>
<tr>
<td>008</td>
<td></td>
</tr>
<tr>
<td>009 6</td>
<td></td>
</tr>
<tr>
<td>010 9</td>
<td></td>
</tr>
</tbody>
</table>

If you limit PROD.NO, you also limit DEL.DATE. However, if you limit PROD.NO, you can still display all the dates by specifying 6 rather than DEL.DATE. The following example shows the difference.
```plaintext
:SORT INVOICE BY-EXP PROD.NO PROD.NO > "7055" DEL.DATE  
limits the display of product numbers;  
the delivery date is displayed only if the 
product number is.

PAGE 1  12:11:59  09 JUL 1992
INVOICE... Product... Delivery.  
Number  Date  
1681  
1683  
1681  7056  08/14/92  
1686  7056  08/10/92  
1687  7056  05/16/92  
1681  7065  08/12/92  
1686  7065  08/12/92  

52 items listed.

:SORT INVOICE BY-EXP PROD.NO PROD.NO > "7055" 6  
limits the display of product numbers;  
the delivery date is always displayed.

PAGE 1  12:11:59  09 JUL 1992
INVOICE... Product... Delivery.  
Number  Date  
1681  
1683  
1681  7056  08/14/92  
1686  7056  08/10/92  
1687  7056  05/16/92  
1681  7065  08/12/92  
1686  7065  08/12/92  

52 items listed.
```
<table>
<thead>
<tr>
<th>Product Number</th>
<th>Delivery Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1681</td>
<td>08/20/92</td>
</tr>
<tr>
<td>1682</td>
<td>08/14/92</td>
</tr>
<tr>
<td>1683</td>
<td>08/12/92</td>
</tr>
<tr>
<td>1686</td>
<td>08/18/92</td>
</tr>
<tr>
<td>1687</td>
<td>08/18/92</td>
</tr>
<tr>
<td>1696</td>
<td></td>
</tr>
</tbody>
</table>

6 items listed.

Use 6 to display the delivery date even if the product number is not specified.
Hints

Suppressing Display of Data

The display of data can be suppressed in the following ways:

- suppress data for an attribute on detail line, but not break line.
- suppress data for an attribute on detail line and break line.
- suppress data for all attributes on detail lines.

Suppressing Data on Detail Lines

The data associated with an attribute can be suppressed on detail lines by setting up an attribute definition item with the following:

- attribute 3 (S/NAME) \ (backslash)
- attribute 10 (V/MAX) 0 (zero)

This does not, however, suppress printing on a break line or total line. You should, however, specify the attribute last, or it may be overwritten and not displayed even on break and total lines.

```
NO-DET
C01 A
C02 34
C03 
C04
C05
C06
C07 \S/NAME\nC08 \V/MAX\nC09 L
C10 C

:SORT INVOICE BY DATE DATE TOTAL NO-DET

PAGE 1
1992-06-01 09 JUL 1992

INVOICE... (N/A)...
Date
1.687 10.84 34
1.696 17
1.683 11
1.686 35
1.681 11.89
1.682 34.92
...
0.00 0.00
6 items listed.
```
Suppressing Display of Data

Suppressing Data on Detail Lines and Break Lines

To suppress printing of an attribute on a break line or total line as well as detail lines, set up V/CONV (attribute 7) to always return a null value.

For example, you can use the following, which concatenates a null:

```
007 C;"
```

The following example shows an attribute definition item that is used to force a break and compile a sub-total after every ten detail lines. The BY-EXP connective causes each product number to be counted as an item. The " " following the BREAK-ON modifier is used to prevent the default asterisks from printing on the break lines.

```
:SORT INVOICE BY-EXP PROD.NO PROD.NO TOTAL EXT.PRICE
BREAK-ON TEN " " ID-SUPP
```

```
Product... Ext. Price
Number
0001 220.74
0002 220.25
0003 220.75
2022 53.00
2023 98.34
2073 50.00
3024 17.61
3025 14.00
3030 17.61

1148.27
8123 020.36
8123 010.30

52 items listed.
```
### Suppressing All Details

All detail lines can be suppressed from a report by using the DET-SUPP modifier in the command. Only the break lines (if any) and grand-total line are output. If there are break lines, the value of the break attribute (up to a limit of 48 characters) is displayed on the lines.

```plaintext
:LIST INVOICE TOTAL EXT.PRICE DET-SUPP

PAGE 1

invoiCE... EXt.PRICE.

*** 517.40
6 items listed.

:sort invoice by COMPANY break-on COMPANY TOTAL EXT.PRICE DET-SUPP

PAGE 1

INVOICE... COMPANY Name........ EXT. PRICE.

Kelley Brothers 538.83
Quality Lighting Prod 6220.93
Service Office Prod 5286.80
Universal Makers 655.83

*** 517.40
6 items listed.
```
BREAK-ON and Forms

The BREAK-ON modifiers force a control break with totals at specified locations in forms. The following technique can be used to force a break after a certain number of forms have been printed:

1. Define a special attribute definition item that uses the NI (number of items) counter in either an A or F correlative or conversion code.

2. Specify the desired number of items to output before each break in the A or F code.

3. Then use that attribute name with a BREAK-ON modifier.

For example, if a break is desired every tenth form, the following attribute definition item would return a changed value every tenth form:

```
TEN
001 A
002 23
003 ...
007 C:""
008 A:(NI)/"10"
009 L
010 0
```

You could then specify a BREAK-ON modifier similar to the following to cause a break every tenth form:

```
BREAK-ON @(3,3):TEN
```

This attribute will not print on detail lines (attribute 10 is 0) or on total lines.
Hints

Testing for Existence

If you want to test for the existence of a specific attribute, that is, the attribute is not null, you can specify the following:

```
SELECT INVOICE WITH NAME
```

Do not specify > "" in an existence test. For left justified attributes, this test always fails.
Several processing codes can be used to verify aspects of data before it is output. The following lists the processing codes; for a complete description of each, see Chapter 5.

A arithmetic code. The IF operator can be set up for a variety of checks.
L length checking. Checks that the number of characters in an attribute fall within minimum and maximum values.
P pattern matching. Checks that data fits a certain pattern.
R range checking. Checks that value of data falls within minimum and maximum values.
Tfile file translation. The V subcode can be used to verify the existence of the value in the specified file before reporting on it.
A Glossary

A/AMC  attribute number heading. The number or position of the attribute in the data item.

ABS  system software.

ACC File  accounting history file. Contains account usage data.

access permission  ability to access another account or a file in another account. Access is provided via update and retrieval locks.

account  a collection of related files associated with one user or one function. Each account has a Master Dictionary.

accountname  name of an account. A user logs on the system by entering an accountname.

algebraic notation  method of specifying a function or formula where operator is specified between the operands; algebraic notation is used in A processing codes.

alphanumeric sort  characters are compared left to right, one character at a time, smallest to largest ASCII value.

AMC  attribute mark count: specified location of attribute in item.

attribute  a line of information in an item. Attributes are delimited by attribute marks.

attribute definition item  used to define an attribute: includes information on how the data is calculated and presented.
<table>
<thead>
<tr>
<th><strong>attribute mark</strong></th>
<th>delimiter used to separate attributes; has decimal ASCII value of 254.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>audit file</strong></td>
<td>file used by forms processing C print specification code to store audit trail information.</td>
</tr>
<tr>
<td><strong>base, base FID</strong></td>
<td>file location; first frame ID (FID) of primary storage.</td>
</tr>
<tr>
<td><strong>BASIC</strong></td>
<td>a programming language, name is acronym for Beginner's All-purpose Symbolic Instruction Code.</td>
</tr>
<tr>
<td><strong>BASIC subroutine</strong></td>
<td>BASIC program that has the BASIC statement SUBROUTINE as its first statement. Subroutines can only be executed by being called from another BASIC program or from a B processing code.</td>
</tr>
<tr>
<td><strong>block (tape)</strong></td>
<td>unit of information on a tape.</td>
</tr>
<tr>
<td><strong>braces</strong></td>
<td>punctuation marks {} in syntax definitions that surround optional parameters.</td>
</tr>
<tr>
<td><strong>checksum</strong></td>
<td>value used to ensure that data remains uncorrupted. The checksum calculation in Ultimate RECALL is based on the binary value of each character times a positional value.</td>
</tr>
<tr>
<td><strong>columnar format</strong></td>
<td>report format where the headings and data are displayed across the top of the page. See also non-columnar format.</td>
</tr>
<tr>
<td><strong>complex itemlist</strong></td>
<td>an itemlist that contains relational operators.</td>
</tr>
</tbody>
</table>
connective

elements of Ultimate RECALL that are used to specify formatting, selection criteria, or sort criteria, such as BREAK-ON, WITH, or BY-EXP. Connectives are part of every new account and are kept in the Master Dictionary.

control break

occurs whenever there is a change in the value of the specified attribute.

controlling attribute

attribute that controls the display of specified attribute; if the controlling attribute is displayed, its dependent attributes are also displayed. Controlling attributes are defined in attribute 4 of an attribute definition item.

controlling/dependent relationship

a set of attributes that are listed together or associated for other Ultimate RECALL purposes.

conversion

processing code that is applied to data in an intermediate format (or in stored format if no correlative were applied) to change it to the external format. Conversions are defined in attribute 7 of an attribute definition item.

correlative

processing code that is applied to stored data to change it to an intermediate format. Correlatives are defined in attribute 8 of an attribute definition item.

D-pointer

see file definition item.

D/CODE

definition code: used to determine type of item. The D/CODE of a file definition item is a D. The D/CODE of an attribute definition item is an A. The D/CODE of a synonym file definition item is a Q.

DATA section

lowest level of a file: contains data stored in variable length attributes and items.
<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default value</td>
<td>preset value for a parameter if no user input is provided.</td>
</tr>
<tr>
<td>default output</td>
<td>attribute definition items with item-IDs that are consecutively numbered from 1 to ( n ) (the highest consecutive number).</td>
</tr>
<tr>
<td>specifications</td>
<td></td>
</tr>
<tr>
<td>delimiter</td>
<td>separator between elements. In an Ultimate RECALL sentence, spaces are used as delimiters.</td>
</tr>
<tr>
<td>dependent attribute</td>
<td>attribute that depends on the existence of another (controlling) attribute; if the controlling attribute is displayed, its dependent attributes are also displayed. Dependent attributes are defined in attribute 4 of an attribute definition item.</td>
</tr>
<tr>
<td>dictionary</td>
<td>file section that contains attribute definition items and pointers to data files.</td>
</tr>
<tr>
<td>dummy AMC</td>
<td>an AMC that is greater than the highest actual AMC on the file.</td>
</tr>
<tr>
<td>dummy attribute</td>
<td>an attribute with a dummy AMC.</td>
</tr>
<tr>
<td>enter</td>
<td>by convention, this means type input, then press the RETURN key.</td>
</tr>
<tr>
<td>ERRMSG file</td>
<td>file containing system messages.</td>
</tr>
<tr>
<td>error message</td>
<td>system response to incorrect entry or other processing error.</td>
</tr>
<tr>
<td>explicit itemlist</td>
<td>the item-IDs of the items to be included in the report are explicitly named in the Ultimate RECALL statement.</td>
</tr>
<tr>
<td>extended items</td>
<td>items with over 32 Kb characters.</td>
</tr>
<tr>
<td>FID</td>
<td>frame ID or frame number.</td>
</tr>
<tr>
<td>file</td>
<td>contains one or more items.</td>
</tr>
<tr>
<td>file definition item</td>
<td>defines and points to a file in the current account. Also known as a D-pointer.</td>
</tr>
</tbody>
</table>
file synonym definition item points to a file in another account. Also known as a Q-pointer.

filename name of the file; can specify the DICT section, one or more DATA sections, or all sections of a file.

forms expression part of an Ultimate RECALL statement; directs Ultimate RECALL to place an attribute or string of output at a specified row and column on the report page.

frame a unit of disk storage; the number of bytes in a frame is platform-specific.

group a collection of frames. The number of groups per file is determined by the modulo of the file.

hashing algorithm an arithmetic formula that is used to determine the group into which items are placed.

heading line top line of report; default heading shows page number and time and date report was created.

hexadecimal base 16 numbering system.

implicit itemlist the itemlist that is used if the Ultimate RECALL statement has no explicit itemlist. If a select-list is present, it is used; otherwise all items in the file are considered.

index a sorted set of data based on attributes in a file. Provides a permanent, pre-sorted means of accessing items in the indexed file.

input conversion refers to changing values in the Ultimate RECALL statement from external format to intermediate format using any conversions.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intermediate format</td>
<td>Format created by Ultimate RECALL for processing; correlatives are applied to the stored data to change it to the intermediate format.</td>
</tr>
<tr>
<td>Internal format</td>
<td>Format in which data is stored.</td>
</tr>
<tr>
<td>Item</td>
<td>A set of related attributes; also known as a record.</td>
</tr>
<tr>
<td>Item-ID</td>
<td>Name by which an item is identified. Also known as key record.</td>
</tr>
<tr>
<td>Itemlist</td>
<td>Part of an Ultimate RECALL statement that specifies items to be considered.</td>
</tr>
<tr>
<td>Justification</td>
<td>The alignment of data in an item for display or sorting.</td>
</tr>
<tr>
<td>Line</td>
<td>Data line between the terminal and the system.</td>
</tr>
<tr>
<td>Logoff</td>
<td>Ends a work session at a terminal and make it inactive.</td>
</tr>
<tr>
<td>Logon</td>
<td>Starts a work session at a terminal in a specified account.</td>
</tr>
<tr>
<td>Logto</td>
<td>Exits the current account and starts a session in the specified account.</td>
</tr>
<tr>
<td>M/DICT</td>
<td>Heading for the Master Dictionary.</td>
</tr>
<tr>
<td>Master dictionary</td>
<td>Contains the account's verbs, PROCs, and cataloged BASIC programs, as well as D-pointers and Q-pointers.</td>
</tr>
<tr>
<td>MD</td>
<td>Master dictionary.</td>
</tr>
<tr>
<td>Message line</td>
<td>Last line of report; shows the number of items on the report.</td>
</tr>
<tr>
<td>Modifier</td>
<td>Connective that controls the format and layout of a report, such as double-spacing.</td>
</tr>
</tbody>
</table>
Glossary

modulo
number of groups in primary storage; maximum is 16,777,213.

multipaged
in a forms report, data from a single item can be reported on multiple pages.

multivalue
one of several values in an attribute.

NEWAC file
prototype master dictionary file.

non-columnar report
the attribute headings are listed down the side of the report with their respective values immediately to the right, one item at a time. Each attribute is displayed on a separate line. See also columnar format.

numeric sort
right to left, one character at a time, smallest to largest numeric value.

option
one-letter code that follows the Ultimate RECALL statement and controls the format and layout of the report. Must be preceded by a left parenthesis and must be specified after all other parameters.

output conversion
refers to changing data from stored format to intermediate format using any correlatives, and then from intermediate format to output format using any conversions.

output format
format created by Ultimate RECALL for printing or displaying.

output specification
specifies the attributes to be used in the report.

parameter
user-specified input to an Ultimate command statement, such as filename, item-ID, or option.

POINTER-FILE
file used by the SAVE-LIST command to store select lists.

print
output at the printer.
<table>
<thead>
<tr>
<th>term</th>
<th>definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>print limiter</td>
<td>indicates criteria that a value in the specified attribute must meet in order to be included on the report.</td>
</tr>
<tr>
<td>PROC</td>
<td>executable program written in Ultimate PROC procedural language. Can be used as an Ultimate command.</td>
</tr>
<tr>
<td>processing code</td>
<td>converts data from internal (stored) format to intermediate or output format and is specified in attribute 7 or 8 of an attribute definition item or file definition item.</td>
</tr>
<tr>
<td>processing format</td>
<td>another name for intermediate format; called processing because both stored and user-supplied values are converted to it before most processing takes place.</td>
</tr>
<tr>
<td>Q-pointer</td>
<td>see file synonym definition item.</td>
</tr>
<tr>
<td>record (tape)</td>
<td>see block.</td>
</tr>
<tr>
<td>relational operator</td>
<td>connective used to test relationships between elements. For example, = or NOT are relational operators.</td>
</tr>
<tr>
<td>report body</td>
<td>contains the detail lines of the report as specified in the statement.</td>
</tr>
<tr>
<td>RETURN</td>
<td>the keyboard RETURN or ENTER key.</td>
</tr>
<tr>
<td>reverse Polish</td>
<td>method of specifying a function or formula where operands are specified before the operator; reverse Polish notation is used in F processing codes.</td>
</tr>
<tr>
<td>notation</td>
<td></td>
</tr>
<tr>
<td>S/AMC</td>
<td>report heading for structure code if the attribute controls or is controlled by another attribute.</td>
</tr>
<tr>
<td>S/NAME</td>
<td>specifies column heading.</td>
</tr>
<tr>
<td>select list</td>
<td>a list of items created with one of the list commands.</td>
</tr>
</tbody>
</table>
Glossary

**selection criteria**
provides the tests that the items chosen for consideration must pass in order to be included in the Ultimate RECALL report.

**separation**
number of contiguous frames per group.

**sort criteria**
used to specify the attribute names to use as the sort keys.

**soundex**
a means of reducing various letter combinations to a common set of sounds, thus allowing comparisons, regardless of upper and lower case letters or slight variations in spelling.

**spooler**
process that controls and routes print jobs.

**statement**
the Ultimate RECALL command and any parameters. Executed by pressing the RETURN key.

**subvalue**
subdivision of a value. Subvalues are delimited by subvalue marks.

**subvalue mark**
delimiter used to separate subvalues; has decimal ASCII value of 252.

**SYSPROG**
system programmer account.

**system command**
any command that can be specified from the TCL level or used anywhere that a TCL level command is valid.

**TAC**
technical assistance center.

**TCL**
terminal control language.

**TCL prompt**
a colon (:) if the TCL stack is ON; otherwise a greater-than sign (>).

**terminal**
serial I/O device with keyboard and monitor screen on which system input and output is displayed.
Appendix A

throwaway modifiers  modifiers that have no effect in an Ultimate RECALL sentence, which "throws them away." These modifiers are included in the standard Ultimate RECALL vocabulary to enhance the English-like syntax of Ultimate RECALL commands.

UltiKit  application development environment.

Ultimate RECALL  a general-purpose data retrieval language that enables you to selectively retrieve information from your database and create customized reports.

Ultimate UPDATE  screen-oriented, online database maintenance functions that allow you to set up data entry screens and use them for updating database files.

UltiPlot  graphics display functions.

V/CONV  contains processing codes that Ultimate RECALL uses to convert data to output format.

V/CORR  contains processing codes that Ultimate RECALL uses to convert data from its internal (stored) format to processing format.

V/MAX  defines the number of characters to display on a line.

V/TYP  defines both the placement of the output (left- or right-justified) in columnar mode and the type of sort used when the file is sorted by this attribute.

value  subdivision of an attribute. Values are delimited by value marks.

value mark  delimiter used to separate values; has decimal ASCII value of 253.
verb
TCL command written in assembly language.

VMC
value mark count; specified location of value in attribute.

wild card
character that stands in for any character in a string. Wild cards are used to specify a partial search string.

window
a portion of a forms page used for displaying multivalued sets of attributes.

X'nn'
hexadecimal value.
Appendix A

Notes
B Default Attribute Definition Items

When an account is created, several attribute definition items are included in the master dictionary of the new account. These items can be used to display or sort attributes in any file in the account.

Table B-1 lists the attributes.
### Table B-1. Default Attribute Definition Items (1 of 2)

<table>
<thead>
<tr>
<th>MD</th>
<th>D/CD</th>
<th>AMC</th>
<th>S/NAME</th>
<th>STRUCT</th>
<th>CONVERSION</th>
<th>Correlative</th>
<th>TP MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;A0&quot;</td>
<td>A</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>10</td>
</tr>
<tr>
<td>&quot;A1&quot;</td>
<td>A</td>
<td>01</td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>10</td>
</tr>
<tr>
<td>&quot;A10&quot;</td>
<td>A</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>10</td>
</tr>
<tr>
<td>&quot;A11&quot;</td>
<td>A</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>10</td>
</tr>
<tr>
<td>&quot;A12&quot;</td>
<td>A</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>10</td>
</tr>
<tr>
<td>&quot;A13&quot;</td>
<td>A</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>10</td>
</tr>
<tr>
<td>*A14&quot;</td>
<td>A</td>
<td>14</td>
<td>Default</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;A15&quot;</td>
<td>A</td>
<td>0</td>
<td>Help</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;A16&quot;</td>
<td>A</td>
<td>16</td>
<td>Links</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;A17&quot;</td>
<td>A</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;A18&quot;</td>
<td>A</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;A2&quot;</td>
<td>A</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;A3&quot;</td>
<td>A</td>
<td>03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*A3DV</td>
<td>A</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;A4&quot;</td>
<td>A</td>
<td>04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*A4DV</td>
<td>A</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;A5&quot;</td>
<td>A</td>
<td>05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;A6&quot;</td>
<td>A</td>
<td>06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*A6DV</td>
<td>A</td>
<td>06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;A7&quot;</td>
<td>A</td>
<td>07</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;A8&quot;</td>
<td>A</td>
<td>08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;A9&quot;</td>
<td>A</td>
<td>09</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*A9998&quot;</td>
<td>A</td>
<td>9998</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*A9999&quot;</td>
<td>A</td>
<td>9999</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*A9DV&quot;</td>
<td>A</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>A</td>
<td>0</td>
<td>NAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>A</td>
<td>1</td>
<td>D/CD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>12</td>
<td>MAX</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>2</td>
<td>AMC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>03</td>
<td>S/VAVL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>A</td>
<td>04</td>
<td>STRUCT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>A</td>
<td>07</td>
<td>CONVERSION</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>A</td>
<td>06</td>
<td>CORRELATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>A</td>
<td>09</td>
<td>TP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>AMC</td>
<td>02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>A</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>A</td>
<td>02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td>A</td>
<td>03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td>A</td>
<td>04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td>A</td>
<td>05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>CODE</td>
<td>1</td>
<td>CODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CODE1</td>
<td>1</td>
<td>CODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>SIZE</td>
<td>9999</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>BASE</td>
<td>02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>LINKS</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>MOD</td>
<td>03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table B-1. Default Attribute Definition Items (2 of 2)

<table>
<thead>
<tr>
<th>MD ..........</th>
<th>D/CD</th>
<th>AMC</th>
<th>S/NAME</th>
<th>STRUCT</th>
<th>CONVERSION</th>
<th>CORRELATIVE</th>
<th>TP</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>F/REALLOC</td>
<td>A</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td>F/SEP</td>
<td>A</td>
<td>04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td>8</td>
</tr>
<tr>
<td>L/RET</td>
<td>A</td>
<td>05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>5</td>
</tr>
<tr>
<td>L/UPD</td>
<td>A</td>
<td>06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>5</td>
</tr>
<tr>
<td>S/AMC</td>
<td>A</td>
<td>04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>2</td>
</tr>
<tr>
<td>S/NAME</td>
<td>A</td>
<td>03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>20</td>
</tr>
<tr>
<td>V/CONV</td>
<td>A</td>
<td>07</td>
<td>CONVERSION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>V/CORR</td>
<td>A</td>
<td>08</td>
<td>CORRELATIVE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>V/EDIT</td>
<td>A</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>10</td>
</tr>
<tr>
<td>V/MAX</td>
<td>A</td>
<td>10</td>
<td>MAX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>V/MIN</td>
<td>A</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>V/TYP</td>
<td>A</td>
<td>09</td>
<td>TP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

58 items listed.
C  Database Used in Examples

The database used in the examples consists of four files:

• CUSTOMERS
• INVOICE
• PROD.NO
• ZIP

The INVOICE file is the basic file from which most of the reports are made. The PROD.NO file is used mainly to translate the product number and return the descriptive name and price per unit. The ZIP file is used mainly to translate the zip code and return the city and state. The CUSTOMERS file was created to demonstrate how REFORMAT and SREFORMAT can be used.

One additional file called ASSEMBLY was used for the examples of the WITHIN modifier and the V correlative.
Attribute 8 (V/CORR) specifies attribute 2 in the ASSEMBLY file as the field that contains the key to the product structure.

```plaintext
:CT DICT ASSEMBLY *

<table>
<thead>
<tr>
<th>ASSEMBLY</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>001 D</td>
<td>001 A</td>
<td>001 A</td>
</tr>
<tr>
<td>002 338861</td>
<td>002 1</td>
<td>002 2</td>
</tr>
<tr>
<td>003 3</td>
<td>003 DESCRIPTION 003 SUB-ASSEMBLY .</td>
<td></td>
</tr>
<tr>
<td>004 1</td>
<td>004</td>
<td>004</td>
</tr>
<tr>
<td>005</td>
<td>005</td>
<td>005</td>
</tr>
<tr>
<td>006</td>
<td>006</td>
<td>006</td>
</tr>
<tr>
<td>007</td>
<td>007</td>
<td>007</td>
</tr>
<tr>
<td>008 V::2</td>
<td>008</td>
<td>008</td>
</tr>
<tr>
<td>009 L</td>
<td>009 T</td>
<td>009 L</td>
</tr>
<tr>
<td>010 10</td>
<td>010 20</td>
<td>010 15</td>
</tr>
</tbody>
</table>

:CT ASSEMBLY *

A100
001 Finished product
A100 is made up of three sub-items: A10, A11, and A12
002 A10|A11|A12
A10
001 Component A
002
A11
001 Component B
A11 is made up of two sub-items: A13 and A14.
002 A13|A14|
A12
001 Component C
002
A13
001 Raw material for B
002
A14
001 Raw material for B
002
```
CUSTOMERS File

The CUSTOMERS file was created by using the REFORMAT command with the INVOICE file and extracting the details for a CUSTOMERS file. The file was also used in the examples for LIST-LABEL and SORT-LABEL.

DATA Section

The following items make up the data section of the CUSTOMERS file.

1
001 Quality Lighting Products
002 Mat
003 5454 W. Mariposa
004 Downey
005 CA
006 91710
007 3015551234

2
001 Service Office Products
002 Lee
003 4512 Orange
004 Santa Ana
005 CA
006 92222
007 7145551234

3
001 Universal Copiers
002 Marina
003 211 Westgate
004 Long Beach
005 CA
006 91832
007 2135551234

4
001 Kelley Brothers
002 Jerry
003 12345 Main Street
004 Anaheim
005 CA
006 92006
007 7145556789

5
001 Service Office Products
002 Shelby
**DICT Section**

The following items make up the dictionary section of the CUSTOMERS file.

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>001 A</td>
<td>001 A</td>
<td>001 A</td>
</tr>
<tr>
<td>002 1</td>
<td>002 2</td>
<td>002 3</td>
</tr>
<tr>
<td>003 Company</td>
<td>003 Contact</td>
<td>003 Address</td>
</tr>
<tr>
<td>004</td>
<td>004</td>
<td>004</td>
</tr>
<tr>
<td>005</td>
<td>005</td>
<td>005</td>
</tr>
<tr>
<td>006</td>
<td>006</td>
<td>006</td>
</tr>
<tr>
<td>007 MCT</td>
<td>007</td>
<td>007</td>
</tr>
<tr>
<td>008</td>
<td>008</td>
<td>008</td>
</tr>
<tr>
<td>009 L</td>
<td>009 L</td>
<td>009 L</td>
</tr>
<tr>
<td>010 20</td>
<td>010 20</td>
<td>010 20</td>
</tr>
<tr>
<td>001 A</td>
<td>001 X</td>
<td>001 X</td>
</tr>
<tr>
<td>002 4</td>
<td>002 5</td>
<td>002 6</td>
</tr>
<tr>
<td>003 City</td>
<td>003 State</td>
<td>003 ZIP</td>
</tr>
<tr>
<td>004</td>
<td>004</td>
<td>004</td>
</tr>
<tr>
<td>005</td>
<td>005</td>
<td>005</td>
</tr>
<tr>
<td>006</td>
<td>006</td>
<td>006</td>
</tr>
<tr>
<td>007 C4, ' ;5 6</td>
<td>007</td>
<td>007</td>
</tr>
<tr>
<td>008</td>
<td>008</td>
<td>008</td>
</tr>
<tr>
<td>009 L</td>
<td>009 L</td>
<td>009 L</td>
</tr>
<tr>
<td>010 20</td>
<td>010 2</td>
<td>010 5</td>
</tr>
<tr>
<td>001 A</td>
<td>001 A</td>
<td>001 A</td>
</tr>
<tr>
<td>002 7</td>
<td>002 3</td>
<td>002 4</td>
</tr>
<tr>
<td>003 Phone Number</td>
<td>003 Address</td>
<td>003 City</td>
</tr>
<tr>
<td>004</td>
<td>004</td>
<td>004</td>
</tr>
<tr>
<td>005</td>
<td>005</td>
<td>005</td>
</tr>
<tr>
<td>006</td>
<td>006</td>
<td>006</td>
</tr>
<tr>
<td>007 MR(###/###)</td>
<td>007 ML(#20)</td>
<td>007 ML(#20)</td>
</tr>
<tr>
<td>008</td>
<td>008</td>
<td>008</td>
</tr>
<tr>
<td>009 R</td>
<td>009 L</td>
<td>009 L</td>
</tr>
<tr>
<td>010 12</td>
<td>010 20</td>
<td>010 20</td>
</tr>
</tbody>
</table>
### CUSTOMERS File

<table>
<thead>
<tr>
<th>COMPANY</th>
<th>NAME</th>
<th>PHONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>001 A</td>
<td>001 A</td>
<td>001 A</td>
</tr>
<tr>
<td>002 1</td>
<td>002 2</td>
<td>002 7</td>
</tr>
<tr>
<td>003 Company Name</td>
<td>003 Contact</td>
<td>003</td>
</tr>
<tr>
<td>004</td>
<td>004</td>
<td>004</td>
</tr>
<tr>
<td>005</td>
<td>005</td>
<td>005</td>
</tr>
<tr>
<td>006</td>
<td>006</td>
<td>006</td>
</tr>
<tr>
<td>007 ML(#30)</td>
<td>007 ML(#20)</td>
<td>007 MR(####/###)</td>
</tr>
<tr>
<td>008</td>
<td>008</td>
<td>008</td>
</tr>
<tr>
<td>009 L</td>
<td>009 L</td>
<td>009 R</td>
</tr>
<tr>
<td>010 30</td>
<td>010 20</td>
<td>010 12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STATE</th>
<th>ZIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>001 A</td>
<td>001 A</td>
</tr>
<tr>
<td>002 5</td>
<td>002 6</td>
</tr>
<tr>
<td>003 State</td>
<td>003 ZIP</td>
</tr>
<tr>
<td>004</td>
<td>004</td>
</tr>
<tr>
<td>005</td>
<td>005</td>
</tr>
<tr>
<td>006</td>
<td>006</td>
</tr>
<tr>
<td>007 ML(#5)</td>
<td>007 MR(#5)</td>
</tr>
<tr>
<td>008</td>
<td>008</td>
</tr>
<tr>
<td>009 L</td>
<td>009 L</td>
</tr>
<tr>
<td>010 5</td>
<td>010 5</td>
</tr>
</tbody>
</table>
INVOICE File

The INVOICE file is used throughout the manual in examples that illustrate features of Ultimate RECALL.

DATA Section

The following items make up the stored data in the INVOICE file. Use the attribute definition items to determine what each attribute actually contains. Also note that many of the data attributes are defined in more than one way.

```
1681
001 Kelley Brothers
002 9002
003 7055|7056|7065|7066|7345|7331|9123|7001|7015|7017|
6032|6068|0005|2025|8036
004 2|1|1|1|2|1|1|1|1|4|1|1|1|6
005 8999|8993|8991|8997|8997|8997|8997|8997|8997|8997|
8997|8997|8997|8997|8997|8997
006 0|0|0|0|0|0|0|0|0|0|0|0|0|0
007 Jerry
008 12345 Main Street
009 920061234
010 Net 30
011 Truck
012 7145556789

1682
001 Kelley Brothers
002 9003
003 7055|6032|6065|e068|5011|5015
004 2|1|1|1|1|1|1
005 8992|8997|8997|8997|8997|8997|8997|8997|8997|8997|
8997|8997|8997|8997|8997|8997
006 0|0|0|0|0|0|0
007 Jerry
008 12345 Main Street
009 92006
010 Net 30
011 Truck
012 7145556789

1683
001 Service Office Products
002 8998
003 7345|7331|8123|6132|6068|0005|2025|9036|4241|5003|
5005
004 3|1|3|1|1|1|1|2|4|1|1|1|1
005 8992|8992|8992|8992|8992|8992|8992|8992|8992|8992|
8992
006 C
```
<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>001 Quality Lighting Products</td>
<td>8185551234</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>002 9034</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>003 7056</td>
<td>7055</td>
<td>7066</td>
<td>7065</td>
<td>2025</td>
</tr>
<tr>
<td>4</td>
<td>004 8989</td>
<td>8989</td>
<td>8989</td>
<td>8989</td>
<td>8989</td>
</tr>
<tr>
<td>5</td>
<td>005 5454 W. Mariposa</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>006 8934</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>007 8934</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>008 4512 Orange</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>009 8949</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>010 8949</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>011 8949</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>012 8949</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>001 Service Office Products</td>
<td>8185551234</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>002 8934</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>003 7056</td>
<td>7055</td>
<td>7066</td>
<td>7065</td>
<td>2025</td>
</tr>
<tr>
<td>16</td>
<td>004 8903</td>
<td>8903</td>
<td>8903</td>
<td>8903</td>
<td>8903</td>
</tr>
</tbody>
</table>
Appendix C

DICT Section

The following attribute definition items are used to format and process the INVOICE file data. The items are sorted by the AMC in attribute 2.

```
0
001 A
002 0
003 Invoice|Number
004
005
006
007
008
009 L
010 10

AVERAGE
001 A
002 0
003 Average Owed
004
005
006
007 A:LPV/ND;MR2S(#12)
008 A*3(TPROD.NO;X;2;2)
009 R
010 12

F.AVERAGE
001 A
002 0
003 Average Owed
004
005
006
007 F:LPV;ND://MR2S(#12)
008 F;4;3(TPROD.NO;X;2;2);*
009 R
010 12

INVOICE#
001 A
002 0
003 Invoice Number
004
005
006
007
008
009 L
010 10
```
DATE

001 A
002 2
003 Invoice Date

004
005
006
007 D2/
008
009 L
010 9

3
01 X
002 3
003 Product Number

004
005
006
007
008
009 L
010 9

4
01 A
002 3
003 Description

004
005
006
007
008
009 L
010 10

PRICE L

001 A
002 3
003 Price

004
005
006
007
008
009 L
010 25

PROD.NO

001 A
002 3
003 Product Number

004 C:DESC:PRICE:QTY:DEL.DATE
005
006
007
008
009 L
010 10

5
001 A
002 4
003 Qty
004
005
006
007
008
009 R
010 4

QTY

001 A
002 4
003 Qty
004 D:PROD.NO
005
006
007
008
009 R
010 4

QTY L

001 A
002 4
003 Qty
004
005
006
007
008
009 R
010 4

6
001 A
002 3
003 Delivery Date
004
005
006
007 12/
004
005
006
007
008 A9["1","5"](TZIP:X;2;2)
009 L
010 2

STATE.L
001 A
002 9
003 State
004
005
006
007 ML(#2)
008 A9["1","5"](TZIP:X;2;2)
009 L
010 2

VERIFY.ZIP
001 A
002 9
003
004
005
006
007
008 A9["1","5"](TZIP;V:1:1)
009 L
010 20

ZIP
001 A
002 9
003 Zip
004
005
006
007 A IF 9(L0) = '9' THEN 9(ML#5-#4) ELSE 9(ML#5)
008
009 L
010 10

ZIP.L
001 A
002 9
003 Zip
004
005
006

TERM
001 A
002 10
003 Terms
004
005
006
007
008
009 L
010 12

VIA
001 A
002 11
003 Ship Method
004
005
006
007
008
009 L
010 12

PHONE
001 A
002 12
003 Phone Number
004
005
006
007 HR(##/####)
008
009 P
010 12

NO-SET
001 A
002 74
003
004
005
006
007 HR#(#10)
008 A*3(TPROD.NO:X:2:2)
009 L
010 C
Appendix C

TEN
001 A
002 34
003 \n004 005
006 007 C;""
008 A:(NI)/"10"
009 L
010 0

MESSAGE
001 A
002 50
003 0
004 005
006 007
008 009 L
010 20

CUMULATIVE
001 A
002 90
003 \n004 005
006 007
008 009 L
010 10

EXT.PRICE
001 A
002 91
003 Ext Price
004 005
006 007
008 009 L
010 10

NEXT.PRICE
001 A
002 91
003 Ext Price
004 005
006 007
008 009 L
010 10

TOTAL.MS2S(#10)
008 A4*3(TPROD.NO;X:2;2)
009 L
010 0

TOTAL_PRICE
001 A
002 90
003 Tot Price
004 005
006 007
008 009 U
010 15

MR2S(#10)
008 A4*3(TPROD.NO;X:2;2)
009 R
010 10

DESC.
001 A
002 991
003 Description
004 E:PROD.NO

DESC.
001 A
002 991
003 Description
004 E:PROD.NO

MR2S(#25)
007 MCT
008 F:3(TPROD.NO;C;1;1)
009 U
010 15

C-12 Itimate RECALL User's Guide
Confidential and Proprietary to The Ultimate Corp.
008 F:3(TPROD.NO;C;1;1)
009 L
010 25

PRICE
001 A
002 992
003 Price
004 D:PROD.NO
005
006
007 MD2
008 F:3(TPROD.NO;C;2;2)
009 R
010 10
Appendix C

PROD.NO File

The product number file contains the descriptive name for each product code and the price per unit. It is used mainly for translations in the INVOICE file.

DATA Section

The following items make up the stored data in the PROD.NO file. Attribute 1 contains the description; attribute 2, the price per unit.

0005
001 Herb Tea
002 2025

2025
001 Regular Tea
002 305

3004
001 Non-dairy Creamer 18/50 Ct
002 2995

3007
001 Sugar Packets 1000 Ct
002 1750

3035
001 Artificial Sweetener 1250
002 750

4241
001 Candy Bars
002 3500

5003
001 Peanuts
002 2320

5005
001 Sunflower Seeds
002 1200

5011
001 Hot Chocolate Regular
002 735

5015
001 Hot Chocolate Sugar Free
002 560
6024
001 Grape Juice
002 1030

6032
001 Orange Juice
002 1540

6055
001 Cranberry Juice
002 1280

6068
001 Grapefruit Juice
002 1280

7001
001 Water
002 1110

7015
001 Raspberry Soda
002 1040

7017
001 Wild Cherry Soda
002 1040

7055
001 Diet Cola
002 960

7056
001 Regular Cola
002 995

7065
001 Diet Root Beer
002 995

7066
001 Root Beer
002 995

7331
001 Tomato Juice
002 980

7345
001 Apple Juice
002 1540
The following items make up the dictionary section of the PROD.NO file.

```
1
  001 A
  002 L
  003 Description
  004
  005
  006
  007 MCT
  008
  009 L
  010 2C

2
  001 A
  002 2
  003 Price
  004
  005
  006
  007 MR2S
  008
  009 R
  010 1C
```

It is important to note that the terms and their corresponding values are listed in a structured format, where each term is associated with a value. This structured data is crucial for the retrieval and processing of information in the PROD.NO file.
ZIP File

The ZIP file uses zip codes as item-IDs. It is used mainly for translations in the INVOICE file.

DATA Section

The following items make up the stored data in the ZIP file. Attribute 1 in each item is the city; attribute 2, the state.

- 91340
  001 San Fernando
  002 CA

- 91710
  001 Downey
  002 CA

- 91832
  001 Long Beach
  002 CA

- 92006
  001 Anaheim
  002 CA

- 92222
  001 Santa Ana
  002 CA

- 92705
  001
  002 CA

- 92714
  001 Irvine
  002 CA

- 97216
  001 Irvine
  002 CA
Appendix C

Notes
Index

^ (wild card) 2-9
# operator 2-5
& operator 2-5
=, <, > (operators) 2-5
@SENONSENENCE (BASIC statement) 5-14
[ , ] (wild cards) 2-9
\ (suppress heading) 4-6

- A -
A (all pages) print specification code 6-8
A - arithmetic code 5-6 thru 5-12
A modifier 2-50
A option
forms alignment 6-17
LIST command 3-11
prompt 6-30
SORT command 3-31
summary 2-52
accessing items 2-4
AFTER operator 2-5
AMC 4-5
AN modifier 2-50
AND operator 2-4, 2-5, 2-15
ANY modifier 2-50
ARE modifier 2-50
arithmetic functions 5-6 thru 5-12, 5-22 thru 5-30
arithmetic operators 5-7, 5-23
attribute definition items (also see Chapter 4)
default output specifications 2-23, 4-3
format 4-2
attribute mark count (amc) 4-5
attributes
listing 3-14, 3-34
suppressing on break lines 7-9
suppressing on detail lines 7-8
testing for existence 7-12
verifying 7-13
audit file (forms) 6-8, 6-10 thru 6-11
averages 5-11, 5-29

- B -
B - BASIC subroutine call 5-13 thru 5-17
B option
background definition 6-18
LIST command 3-11
prompt 6-30
SORT command 3-31
summary 2-52
BASIC
@SENONSENENCE 5-14
CHAIN 5-14
ENTER 5-14
EXECUTE 5-14
GET function 5-14
select lists 2-14
subroutines 5-13 thru 5-17
SYSTEM(18) 5-14
WRTFET 5-14
writing to files 2-54, 5-14
BEFORE operator 2-5
brackets (wild cards) 2-9
BREAK-ON modifier
as output specification 2-23
at specified locations 7-11
defining 2-30 thru 2-32
in forms 6-15, 6-22
with DET-SUPP 2-34
with FOOTING 2-35
with HEADING 2-39
with Title translation 5-54
with TOTAL 2-45
BY 2-18
BY-DSND 2-18
BY-EXP 2-18
BY-EXP-DSND 2-18
bytes in file 3-9
- C -
C (check) print specification code 6-8, 6-30
C - concatenation code 5-18
C option
forms output 6-17
LIST command 3-11
LIST-LABEL command 3-17
REFORMAT command 3-22
SORT command 3-31
SORT-LABEL command 3-38
SREFORMAT command 3-42
summary 2-52
C.AMC 5-15
C.DATA 5-14, 5-15
C.DICTPRIMFILE 5-14, 5-15
C.FLAGS 5-15
C.ITEM 5-15
C.ITEMID 5-15
C.PRIMFILE 5-14, 5-15
C.PRIMFILENAME 5-15
C.SYMC 5-15
C.USER(l5) 5-15
C.VMC 5-15
caret (wild card) 2-9
CHAIN (BASIC statement) 5-14
CHECK-SUM command 3-4 thru 3-5
COL-HDR-SUPP modifier 2-33, 6-15
column heading
description 1-2, 1-3
in forms 6-15
multiple lines 4-6
specifying 4-5
suppressing 2-32, 4-6
columnar report 1-2
columns
suppressing display 4-8
width 4-8
commands (see Chapter 3)
complex item lists 2-4
components of Ultimate RECALL 1-5
concatenation 5-7, 5-18, 5-23
conditional expressions 5-8
connectives 1-5, 2-3
connectives (table) 1-6 thru 1-8
controlling attributes
defining 4-6 thru 4-7
displaying 1-3
in forms 6-4, 6-21
null values in forms 6-24
output specifications 2-24
print limiters 2-26
conversion codes
in attribute definition item 4-7
defining 5-4
effects on TOTAL 2-45, 5-10, 5-28
LIST-ITEM 3-15
SORT-ITEM 3-35
correlative codes
in attribute definition item 4-7
defining 5-4
effects on TOTAL 2-45, 5-10, 5-28
LIST-ITEM 3-15
SORT-ITEM 3-35
COUNT command 3-6
credit indicator codes 5-36, 5-41
CT command 3-15, 3-35

- D -
D - date code 5-19 thru 5-21
D option
LIST command 3-12
LIST-LABEL command 3-17
SORT command 3-31
SORT-LABEL command 3-38
summary 2-52
D/ CODE 4-5
data formats 5-4 thru 5-5
DATA modifier 2-50
dates
converting 5-19 thru 5-21
current 5-6, 5-22
formats 5-19
sorting 5-20
years 5-20
DBL-SPC modifier 2-33, 6-15
default heading 1-2
default output specifications 2-23, 2-28, 2-45, 4-3
default reports 7-2
DELETE-LIST command 2-13
delimiters 2-2
dependent attributes
  column headings 1-2
  defining 4-6 thru 4-7
  displaying 1-3
  in forms 6-4, 6-21
  null values in forms 6-24
  output specifications 2-24
  print limiters 2-26
DET-SUPP modifier 2-34, 2-45
detail lines
  applying processing codes 5-10, 5-28
  counting 5-6, 5-22
  displaying 1-2
  doublespacing 2-33
  processing flags 5-15
  suppressing 2-34, 2-45, 2-52, 7-8
dummy attributes 4-5, 5-10, 5-28

- E -
EACH 2-15
EDIT-LIST command 2-13
END-WINDOW modifier 6-12
ENTER (BASIC statement) 5-14
EQ operator 2-5
EVERY 2-15
EXECUTE (BASIC statement) 5-14
explicit itemlists 2-12
extended items 3-9
extracting fields 5-31
extracting text 5-52

- F -
F (first page) print specification code 6-9
F - function code 5-22 thru 5-30
F option
  LIST-ITEM command 3-15
  SORT-ITEM command 3-35
  summary 2-52
field extracting code 5-31
file allocation 7-2
file definition item
  dumping 3-26, 3-52
  format 2-11
  V correlative 2-48, 5-57
FILE modifier 2-50
filename 2-1
files 2-11
FOOTING modifier
  defining 2-35 thru 2-36
  in forms 6-16
  with BREAK-ON 2-30
FOR modifier 2-50
format masks 5-36 thru 5-38, 5-41 thru 5-43
forms (see Chapter 6)
forms expression 6-5 thru 6-7
forms output 2-3

- G -
G - group code 5-31
GE operator 2-5
GET (BASIC function) 5-14
GET-LIST command 2-13
GRAND-TOTAL modifier 2-37, 6-15
GT operator 2-5

- H -
H option
  LIST command 3-12
  LIST-ITEM command 3-15
  LIST-LABEL command 3-18
  REFORMAT command 3-22
  S-DUMP command 3-26
  SORT command 3-31
  SORT-ITEM command 3-35
  SORT-LABEL command 3-38
  SREFORMAT command 3-42
  summary 2-52, 2-53
  T-DUMP command 3-51
Hash-Test command 3-7 thru 3-8
Hashing algorithm 3-8
HDR-Supp modifier 2-40, 6-16
HEADER modifier (see HEADING)
HEADING modifier
  defining 2-39 thru 2-41
  in forms 6-16
  with BREAK-ON 2-30
Hexadecimal values 5-46

- I -
I option
  LIST command 3-12
  LIST-ITEM command 3-15
  LIST-LABEL command 3-18
  REFORMAT command 3-22
  S-DUMP command 3-26
  SORT command 3-31
  SORT-ITEM command 3-35
  SORT-LABEL command 3-38
  SREFORMAT command 3-42
  summary 2-53
  T-DUMP command 3-51
  T-LOAD command 3-54
ID-Supp modifier 2-42, 6-16
IF modifier 2-15
  implicit itemlists 2-12
IN modifier 2-50
Indexes 7-13
Input conversions 5-4
Intermediate format 5-4 thru 5-5
Internal format 5-4 thru 5-5
ISTAT command 3-7, 3-9 thru 3-10
Item count 3-6, 4-5, 5-7, 5-23
Item size 3-9, 4-5
Item-IDs
  characters not recommended 4-4
  displaying 2-11
  displaying only 2-43
  in footings 2-35
  length 4-3
  referencing 4-5

- J/L -
JULIAN date 5-19
Justification 4-8
L (last page) print specification code 6-9
L - length code 5-32
LE operator 2-5
Left bracket (wild card) 2-9
Length check 5-32
LIKE operator 2-5, 2-7
Limiters
  print 2-2, 2-26 thru 2-27
  sort 2-20
LIST command 3-11 thru 3-13
LIST-ITEM command 3-14 thru 3-16
LIST-LABEL command 3-17 thru 3-20
LPTR 2-44
LT operator 2-5

- M -
M (message) print specification code 6-9
M option (multiple reports)
  description 6-19
  LIST command 3-12
  prompt 6-30
SORT command 3-31
summary 2-53
mailing labels
sorted 3-37 thru 3-40
unsorted 3-17 thru 3-20
mask characters 5-33
mathematical functions 5-22 thru 5-30
MC - mask character code 5-33
MD - mask decimal 5-35
message line 1-3
ML - mask decimal, left justified 5-36 thru 5-38
modifiers 2-2, 2-28 thru 2-50
modulos 3-7
MP - mask packed decimal 5-39 thru 5-40
MR - mask decimal, right justified
description 5-41 thru 5-43
with wild cards 2-9
MT - mask time 5-44 thru 5-45
multi-valued attributes
select lists 2-22
sorting 2-20 thru 2-22
MX - mask hexadecimal 5-46

- N -

N option
HASH-TEST command 3-7
ISTAT command 3-9
LIST command 3-12
LIST-ITEM command 3-15
LIST-LABEL command 3-18
SORT command 3-31
SORT-ITEM command 3-35
SORT-LABEL command 3-38
summary 2-53
ND (detail line) counter
A processing code 5-6
averages 5-11, 5-29
F processing code 5-22
NE operator 2-5
NO operator 2-15
non-columnar report 1-2, 1-4
NOPAGE 2-42

NOT operator 2-5, 2-15
null values
as pad characters 3-22, 3-42
effects on forms 6-24 thru 6-25
in controlling attributes 6-4
in translations 5-53, 6-4
initializing variables 5-13
input conversions 5-44
length test 5-32
pattern test 5-47
printing labels 3-18, 3-39
processing 5-1
range test 5-49
selecting non-null 2-15, 2-16, 7-12
substituting for 5-50
text extraction 5-52
zeros in place of 5-25
number of characters 5-32
number of items 3-6, 4-5, 5-7, 5-22

- O -

O option
summary 2-53
T-LOAD command 3-54
OF modifier 2-50
ONLY modifier 2-23, 2-43, 6-16
operators
arithmetic 5-7, 5-23
order of operations 5-9
relational 2-2, 2-4 thru 2-8, 5-7, 5-24
options 2-2, 2-51 thru 2-54
OR operator 2-4, 2-5, 2-15
order of operations 5-9
output conversions 5-4
output specifications
controlling and dependent attributes 2-24
default 2-23
defining 2-23 thru 2-24
in commands 2-2
in forms 2-3, 6-5 thru 6-7
Index

- P -
P - pattern match code 5-47 thru 5-48
P option 2-53 (also see each command in Chapter 3)
  page length 1-2
  PAGE modifier 2-50
  page width 1-2
  pattern matching 5-47 thru 5-48
  pause at bottom of page 1-2
  POINTER-FILE 2-13
  print limiters 2-2, 2-26 thru 2-27
  print specification codes 6-8 thru 6-11
  printing output 2-42
  processing codes (see Chapter 5 for description of individual processing codes)
    conversions 5-4
    correlatives 5-4
    error checking 5-1
    multiple 5-5

- Q/R -
Q-pointers 2-11
QSELECT command 2-13
R - range code 5-49
  range checking 5-49
  RCL.COMMON 5-13
  REFORMAT command 3-21 thru 3-24
  relational operators
    A processing code 5-7
    defining 2-4 thru 2-8
    effects on item lists 2-12
    F processing code 5-24
    using 2-2
  remainder function 5-7
  report format
    columnar 1-2, 1-4
    default heading 1-2
    forms (see Chapter 6)
    message line 1-3
    non-columnar 1-2, 1-4
    page settings 1-2
    report body 1-2

  reverse Polish notation 5-25
  right bracket (wild card) 2-9

- S -
S - substitution code 5-50
S option
  HASH-TEST command 3-7
  ISTAT command 3-9
  LIST-ITEM command 3-15
  SORT-ITEM command 3-35
  summary 2-53
  S-DUMP command 3-25 thru 3-27
  S/AMC (also see controlling attributes)
    defining 4-6 thru 4-7
    in forms 6-21
  S/NAME 4-5
  SAVE-LIST command 2-13
  select-criteria 2-1
  SELECT command 3-28 thru 3-29
  select lists
    defining 2-13 thru 2-14
    multivalued attributes 2-22
    SELECT command 3-28
    SSELECT command 3-45
    using with other software 2-14
  selection criteria 2-1, 2-15 thru 2-17
  size of item 3-9, 4-5
  skip attribute output 4-5
  slave printers
    with forms 6-30
  SORT command 3-30 thru 3-33
  sort-criteria 2-1
  SORT-ITEM command 3-34 thru 3-36
  SORT-LABEL command 3-37 thru 3-40
  sorting
    commands 2-19
    criteria 2-1, 2-18 thru 2-22
    dates 5-20
    effects of conversions and correlatives 3-32, 5-4
    limiters 2-20
    multivalued attributes 2-20 thru 2-22
    order 2-18

index-6
rules 4-9 thru 4-11
S-DUMP command 3-25 thru 3-27
SORT command 3-30 thru 3-33
SORT-ITEM command 3-34 thru 3-36
SORT-LABEL command 3-37 thru 3-40
SREFORMAT command 3-41 thru 3-44
SSELECT command 3-45 thru 3-46
soundex capability 2-7
SREFORMAT command 3-41 thru 3-44
SSELECT command 3-45 thru 3-46
STAT command 3-47 thru 3-48
structure code 4-6 thru 4-7
subpages (forms) 6-3, 6-12 thru 6-14
subroutines (BASIC) 5-13 thru 5-17
substituting values 5-50
substring function 5-8, 5-23
subtotals 2-31, 2-45, 5-10, 5-28
subvalue count 5-7, 5-23
SUM command 3-49 thru 3-50
summation function 5-8, 5-24
SUPP modifier 2-44
synonym definition item 4-5
SYSLIB 5-13
system commands (see Chapter 3)
    using select lists 2-14
SYSTEM(18) (BASIC statement) 5-14

- T -
T - text extraction code 5-52
T-ATT command
    REFORMAT command 3-23
    S-DUMP command 3-26
    SREFORMAT command 3-45
    T-DUMP command 3-52
    T-LOAD command 3-55
    TAPE modifier 2-44
T-DUMP command 2-44, 3-51 thru 3-53
T-LOAD command 3-54 thru 3-55
TAPE modifier 2-44
tape format
    REFORMAT command 3-23
    S-DUMP command 3-26
SREFORMAT command 3-43
T-DUMP command 3-52
tape labels
    S-DUMP command 3-26
    specifying 2-40
    suppressing 2-33, 2-38, 2-44
T-DUMP command 3-52
tape operations 5-14
text extraction code 5-52
Tfile - translation code
    defining 5-53 thru 5-56
    in forms 6-4, 6-24 thru 6-25
THE modifier 2-50
throwaway modifiers 2-50
time
    converting 5-44 thru 5-45
    current 5-7, 5-23
    formats 5-44
TOTAL modifier
    as output specification 2-23
    creating subtotals 2-45
    defining 2-45 thru 2-46
difference between conversions and correlatives
    5-10, 5-28
    in forms 6-16
    with GRAND-TOTAL 2-37
    with Tfile translations 5-54
    translations 5-53 thru 5-56

- U -
Ultimate RECALL commands (see Chapter 3)
Ultimate UPDATE definition item 4-5
UltiWord
    select lists 2-14
UltiWriter
    select lists 2-14
USING modifier 2-47

- V -
V/CONV 4-7
V/CORR 4-8
V/MAX
Index

- X/Y/Z -

X option
- LIST-ITEM command 3-15
- SORT-ITEM command 3-35
- summary 2-54
- years 5-20

Z option
- LIST command 3-12
- reset page number 6-20
- SORT command 3-31
- summary 2-54

- W -

W option
- BASIC calls 5-14
- LIST command 3-12
- LIST-LABEL command 3-18
- REFORMAT command 3-22
- SELECT command 3-28
- SORT command 3-31
- SORT-LABEL command 3-38
- SREFORMAT command 3-42
- SSELECT command 3-45
- summary 2-54

wild cards
- description 2-9
- in itemlists 2-12
- MR conversions 5-42
- WINDOW modifier 6-12
- windows 6-12 thru 6-14
- WITH connective 2-15
- WITHIN modifier 2-11, 2-48 thru 2-49, 5-57
- WITHOUT connective 2-15
- WRITET (BASIC statement) 5-14
- writing conventions x
# Ultimate

**THE ULTIMATE CORP.**

---

**Problem Identification Form**

<table>
<thead>
<tr>
<th>Name</th>
<th>Phone Number</th>
<th>System Number</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

At TCL, execute REV verb and enter the following information:

- Firmware rev.
- Kernel rev.
- Async rev.
- Abs rev.
- Diags rev.
- ECOs

Hardware Platform: (manufacturer, model no.)

Host O/S and revision

Dealer Name

At TCL, execute WHAT (LSWP) verb and attach listing to this report.

Description of what happened and steps necessary to recreate (attach listings, tapes, if available):
FROM:

Name: ________________ System Number: ____________

Company ________________________________________

Address: _______________________________________

City: __________________ State: ______ Zip: _______

Fold and tape. Please do not staple.

The Ultimate Corp.
717 Ridgedale Avenue
East Hanover, NJ 07936
Attn: Technical Support

Fold and tape. Please do not staple.
Ultimate welcomes your comments. If you find a problem or error in this manual, or can suggest an improvement, please complete this form. Please attach additional sheets, if necessary.

<table>
<thead>
<tr>
<th>Name</th>
<th>Phone Number</th>
<th>System Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name of Manual</th>
<th>Document Number</th>
<th>Date</th>
</tr>
</thead>
</table>

Comments:
Ultimate welcomes your suggestions. If you have a suggestion or would like to recommend an enhancement, please complete this form. Please attach additional sheets, if necessary.

<table>
<thead>
<tr>
<th>Name</th>
<th>Phone Number</th>
<th>System Number</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company Name</td>
<td>Hardware Platform: (manufacturer, model no.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dealer Name</td>
<td>Host O/S and revision</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Suggestion:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fold and tape. Please do not staple.

The Ultimate Corp.
717 Ridgedale Avenue
East Hanover, NJ 07936
Attn: Technical Support

Fold and tape. Please do not staple.