Ultimate

THE ULTIMATE CORP.

RECALL
User's Guide

The Ultimate Corp.
East Hanover, NJ
Version 1.0

Ultimate RECALL® User's Guide
Version 1

Publication Information

Operating System Release 10, Revision 210
©1992, The Ultimate Corp., East Hanover, NJ

UltiWriter
©1989, The Ultimate Corp., East Hanover, NJ and KEO/SOFT International Corporation, Tustin, CA

Document Number 6972

Contents

Index

Figures

Tables

Notes

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 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 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:

Convention	Description
UPPER CASE	Characters in syntax definitions printed in upper case are required and must appear exactly as shown.
lower case	Characters in syntax definitions printed in lower case are parameters to be supplied by the user.
{ }	Braces surrounding a parameter indicate that the parameter is optional and may be included or omitted at the user's option.
RETURN	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.
enter	The word enter is used to mean "type in the required text, then press RETURN."
X'nn'	This form is used to define a hexadecimal number where 'nn' is the hex value; for example, X'0B', X'41', X'FF'.
`Enter option`	This typeface is used for messages and prompts displayed by the system.
CAUDIT	This typeface in bold is used in examples to indicate user input.
filename	Any valid filename format can be used when filename is specified. Filename formats are described in the *Ultimate System Commands Guide.*

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.

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

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.

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.

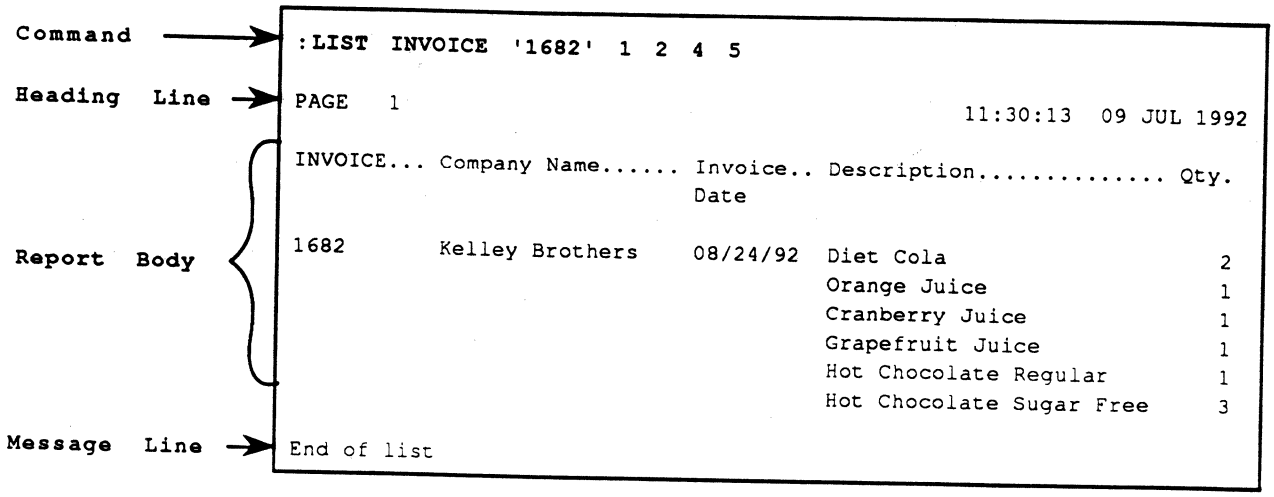


Figure 1-1. Columnar Report Format

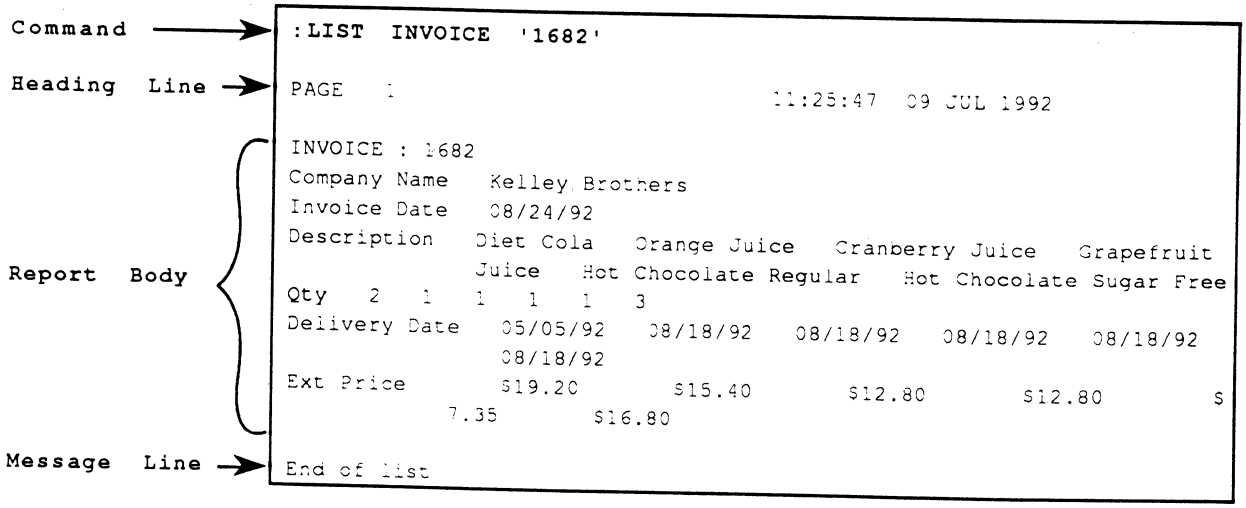


Figure 1-2. Non-Columnar Report Format

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 1-1. Connectives (1 of 3)

Connective	Usage
#	relational operator.
&	relational operator.
<	relational operator.
<=	relational operator.
=	relational operator.
=<	relational operator.
=>	relational operator.
>	relational operator.
>=	relational operator.
A	throwaway modifier.
AFTER	relational operator.
AN	throwaway modifier.
AND	relational operator.
ANY	throwaway modifier.
ARE	throwaway modifier.
BEFORE	relational operator.
BREAK-ON	modifier.
BY	sort criteria.
BY-DSND	sort criteria.
BY-EXP	sort criteria.
BY-EXP-DSND	sort criteria.
COL-HDR-SUPP	modifier.

Table 1-1. Connectives (2 of 3)

Connective	Usage
DATA	throwaway modifier.
DBL-SPC	modifier.
DET-SUPP	modifier.
EACH	selection criteria.
END-WINDOW	forms.
EQ	relational operator.
EVERY	selection criteria.
FILE	throwaway modifier.
FOOTING	modifier.
FOR	throwaway modifier.
GE	relational operator.
GRAND-TOTAL	modifier.
GT	relational operator.
HDR-SUPP	modifier.
HEADER	modifier.
HEADING	modifier.
ID-SUPP	modifier.
IF	selection criteria.
IN	throwaway modifier.
ITEMS	throwaway modifier.
LE	relational operator.
LIKE	relational operator.

Table 1-1. Connectives (3 of 3)

Connective	Usage
LPTR	modifier.
LT	relational operator.
NE	relational operator.
NO	modifier.
NOPAGE	modifier.
NOT	relational operator.
OF	throwaway modifier.
ONLY	modifier.
OR	relational operator.
PAGE	throwaway modifier.
SUPP	modifier.
TAPE	modifier.
THE	throwaway modifier.
TOTAL	modifier.
USING	modifier.
WINDOW	forms.
WITH	selection criteria.
WITHIN	modifier.
WITHOUT	selection criteria.

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

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

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

Command	Description
SREFORMAT	produces a new file from selected items and attributes, sorted by specified attribute values, on disk or tape.
SSELECT	produces a select-list that contains selected item-ids and, optionally, specified attributes, sorted by specified attribute values.
STAT	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.
SUM	outputs the total value of a specified attribute in all selected items.
T-DUMP	outputs selected items to tape.
T-LOAD	loads selected or specified file items from the attached tape.

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 re-sequence 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:

```
WITH NAME < "K" OR > "M" 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:

```
WITH DATE < "07/01/92" > "08/15/92" DATE COMPANY

WITH DATE GE "07/01/92" AND LE "08/15/92" DATE COMPANY
```

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

```
INVOICE < "1685""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.

```
LIST INVOICE = "1682"

LIST INVOICE '1682'
```

Table 2-1. Relational Operators

Operator	Description
EQ =	equal to.
GT > AFTER	greater than.
LT < BEFORE	less than.
GE >= =>	greater than or equal to.
LE <= =<	less than or equal to.
NE # NOT	not equal to.
LIKE	search for items that sound like; for more information, see next subsection.
AND &	both conditions must be true.
OR	one or both conditions must be true.

```
:SORT INVOICE WITH DATE < "07/01/92" > "08/15/92" DATE
COMPANY

PAGE   1                                        16:33:51  13 MAR 1992

INVOICE... Invoice.. Company Name........
           Date
1681       08/23/92  Kelley Brothers
1682       08/24/92  Kelley Brothers
1686       08/21/92  Quality Lighting Products
1687       06/16/92  Service Office Products

4 items listed.


:SORT INVOICE WITH DATE GE "07/01/92" AND LE "08/15/92"
DATE COMPANY

PAGE   1                                        16:33:15  13 MAR 1992

INVOICE... Invoice.. Company Name........
           Date
1683       08/09/92  Service Office Products
1696       07/01/92  Universal Copiers

2 items listed.


:SORT INVOICE < "1685""1696" COMPANY

PAGE   1                                        16:32:18  13 MAR 1992

INVOICE... Company Name........
1681       Kelley Brothers
1682       Kelley Brothers
1683       Service Office Products
1696       Universal Copiers

4 items listed.


:SORT INVOICE WITH NAME < "K" OR > "M" AND < "S" NAME
PAGE   1                                        10:13:29  30 APR 1992

INVOICE... Contact.............
1681       Jerry
1682       Jerry
1686       Mat
1696       Marina

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	

```
:SORT  INVOICE  WITH  NAME  LIKE  "JERI"  COMPANY  NAME

PAGE   1                                          14:02:03  09 JUL 1992

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

1681       Kelley Brothers      Jerry
1682       Kelley Brothers      Jerry

2 items listed.
```

Wild Cards

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 2-2. Wild Cards

Character	Description
...^...	caret; matches any single character.
...]	right bracket; matches all characters to the right of the specified string.
[...	left bracket; matches all characters to the left of the specified string.
[...]	left and right brackets; match all attributes or item-IDs that contain the specified string.

```
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.

Files

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 DICT POINTER-FILE. (Before creating a select-list that you want to save, be sure you have defined a DICT POINTER-FILE for your account.) You can list the stored select-lists for your account by using the following statement:

```
SORT ONLY DICT POINTER-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 AFILE 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 AFILE.

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

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 **IF**	specifies selection criteria; WITH and IF are synonymous and can be used interchangeably.
NO **NOT**	excludes items that have the specified values; NO and NOT are synonymous and can be used interchangeably.
WITHOUT	excludes items that have the specified values. This is equivalent to WITH NO.
EVERY **EACH**	used for multivalued attributes to specify that all subvalues must have the specified values; EVERY and EACH are synonymous and can be used interchangeably.
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 attrib-name
```

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]"

PAGE   1                                          16:44:29  13 MAR 1992

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 =
"O" COMPANY DATE TOT.PRICE
```

sorts all items where the value of the first delivery date is less than 08/01/92.

```
PAGE   1                                          13:21:48  19 JUN 1992

INVOICE... Company Name........ Invoice.. Tot Price.
                                Date

1696       Universal Copiers    07/01/92      $55.80

End of list
```

```
:LIST INVOICE DEL.DATE NAME COMPANY WITH EVERY DEL.DATE
BEFORE "08/01/92"
```

lists invoices with all delivery dates before 08/01/92. (DEL.DATE is dependent on PROD.NO and is not listed, but the items are selected.)

```
PAGE   1                                        13:21:30  09 JUL 1992

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


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

BY attrib-name
BY-DSND attrib-name
BY-EXP attrib-name {exp-limiter}
BY-EXP-DSND attrib-name {exp-limiter}

BY	sorts items according to the ascending value (lowest to highest) of the specified attribute.
BY-DSND	sorts items according to the descending value (highest to lowest) of the specified attribute.
BY-EXP	explodes each multivalue into its own item for the report and sorts in ascending order.
BY-EXP-DSND	explodes each multivalue into its own item for the report and sorts in descending order.
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:

{{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	SPIE
SORT	SPLOT
SORT-ITEM	SREFORMAT
SORT-LABEL	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
```

```
:SORT  INVOICE  COMPANY                    sorts by invoice number and displays
                                           company name.

PAGE   1                                              08:44:17  09 JUL 1992

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

1681       Kelley Brothers
1682       Kelley Brothers
1683       Service Office Products
1686       Quality Lighting Products
1687       Service Office Products
1696       Universal Copiers

6 items listed.

:SORT  INVOICE  BY-DSND  INVOICE#  COMPANY
                                           sorts by descending invoice number.

PAGE   1                                              08:45:41  09 JUL 1992

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

1696       Universal Copiers
1687       Service Office Products
1686       Quality Lighting Products
1683       Service Office Products
1682       Kelley Brothers
1681       Kelley Brothers

6 items listed.
```

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.

```
PAGE   1                                          09:03:29  09 JUL 1992

INVOICE...  Product...  Description....
            Number
                        *
1681        0005        Herb Tea
1681        2025        Regular Tea
1682        5011        Hot Chocolate Regular
1682        5015        Hot Chocolate Sugar Free
1681        6032        Orange Juice
1682        6032        Orange Juice
1682        6065        Cranberry Juice
1681        6068        Grapefruit Juice
1682        6068        Grapefruit Juice
1681        7001        Water
1681        7015        Raspberry Soda
1681        7017        Wild Cherry Soda
1681        7055        Diet Cola
1682        7055        Diet Cola
1681        7056        Regular Cola
1681        7065        Diet Root Beer
1681        7066        Root Beer
1681        7331        Tomato Juice
1681        7345        Apple Juice
1681        8036        Cookies
1681        8123        Crackers

21 items listed.
```

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

limits the explosion to product numbers greater than 8000.

```
PAGE   1                                          09:05:17  09 JUL 1992

INVOICE...  Product...  Description....
            Number
                        *
1681        8036        Cookies
1681        8123        Crackers

2 items listed.
```

BY-EXP and Select Lists

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.
:SAVE-LIST  SSEL

'SSEL' saved - 1 frames used.

:EDIT-LIST  SSEL
Top
.P
001 1681]00D
002 1681]00E
003 1682]005
004 1682]006
005 1681]00B
006 1682]002
007 1682]003
008 1681]00C
009 1682]004
010 1681]008
011 1681]009
012 1681]00A
013 1681]001
014 1682]001
015 1681]002
016 1681]003
017 1681]004
018 1681]006
019 1681]005
020 1681]00F
021 1681]007
EOI 021
```

select list contains the item-IDs and value number of the location of the product number.

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.

```
:SORT  INVOICE                                    Lists all default attributes.

PAGE   1                                                  12:11:51  09 JUL 1992

INVOICE : 1681
Company Name    Kelley Brothers
Invoice Date    08/23/92
Description     Diet Cola    Regular Cola    Diet Root Beer    Root Beer    Apple Jui
                ce    Tomato Juice    Crackers    Water    Raspberry Soda    Wild Cher
                ry Soda    Orange Juice    Grapefruit Juice    Herb Tea    Regular Te
                a    Cookies
Qty    2    1    1    1    2    1    1    1    1    1    4    1    1    1    6
Delivery Date    08/20/92    08/14/92    08/12/92    08/18/92    08/18/92    08/18/92
                    08/18/92    08/18/92    08/18/92    08/18/92    08/18/92    08/18
                 /92    08/18/92    08/18/92    08/18/92
Ext Price         $19.20          $9.95          $9.95          $9.90          $30.80
                 $9.60        $10.30        $11.10        $10.40        $10.40
              $61.60        $12.80        $20.25        $3.05        $40.20
   .
   .
6 items listed.


:SORT  INVOICE  COMPANY.NAME  PHONE      Lists specified attributes.


PAGE   1                                          12:20:58  09 JUL 1992

INVOICE... Company Name........ Contact.............. Phone Number

1681        Kelley Brothers       Jerry                 714/555 6789
1682        Kelley Brothers       Jerry                 714/555 6789
1683        Service Office ProducShelby                 818/555 1234
1686        Quality Lighting ProdMats                   301/555 1234
1687        Service Office ProducLee                    714/555 1234
1696        Universal Copiers     Marina                213/555 1234

6 items listed.
```

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.

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

PAGE   1                                          08:56:34  09 JUL 1992

INVOICE... Company Name..... Product... Qty.
                             Number
                                        *
1682       Kelley Brothers   7055          2
                             6032          1
                             6065          1
                             6068          1
                             5011          1
                             5015          3

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.

```
PAGE   1                                          13:52:58  18 MAR 1992

INVOICE... Phone Number Company Name........

1681       714/555 6789 Kelley Brothers
1682       714/555 6789 Kelley Brothers
1683                    Service Office Products
1686                    Quality Lighting Products
1687       714/555 1234 Service Office Products
1696                    Universal Copiers

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.

```
PAGE   1                                          14:00:11  18 MAR 1992

INVOICE... Product... Description.... Price..... Qty.
           Number
                      *                *          *
1681       6032       Orange Juice        15.40     4
           6068       Grapefruit Juice    12.80     1
1682       6032       Orange Juice        15.40     1
           6065       Cranberry Juice     12.80     1
           6068       Grapefruit Juice    12.80     1
1683       6032       Orange Juice        15.40     1
           6068       Grapefruit Juice    12.80     1
1686       6024       Grape Juice         10.30     1
           6032       Orange Juice        15.40     2
1687
1696

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

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

BREAK-ON

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

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.

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 (***) in the BREAK-ON attribute column, or if optional text has been specified, the text is output instead of the 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.

```
:SORT  INVOICE  BY  VIA  BREAK-ON  VIA  DATE  COMPANY

PAGE   1                                          09:07:24  09 JUL 1992

INVOICE... Ship Method. Invoice.. Company Name........
                        Date

1683       SOP TRUCK    08/09/92  Service Office Products
1687       SOP TRUCK    06/16/92  Service Office Products

           ***

1681       Truck        08/23/92  Kelley Brothers
1682       Truck        08/24/92  Kelley Brothers
1686       Truck        08/21/92  Quality Lighting Products

           ***

1696       UPS          07/01/92  Universal Copiers

           ***

***

6 items listed.
```

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.

```
: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       08/21/92  Quality Lighting Products
1687       06/16/92  Service Office Products
1696       07/01/92  Universal Copiers
```

DBL-SPC

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

```
: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       08/21/92  Quality Lighting Products

1687       06/16/92  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.

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

PAGE   1                                        09:41:28  17 JUN 1992

INVOICE... Ext Price.

***           $917.40

6 items listed.


:SORT  INVOICE  BY  VIA  BREAK-ON  VIA  TOTAL  EXT.PRICE  DET-
SUPP

PAGE   1                                        09:42:33  17 JUN 1992

INVOICE... Ship Method. Ext Price.

           SOP TRUCK       $286.80

           Truck           $574.80

           UPS              $55.80
***                        $917.40

6 items listed.
```

FOOTING

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 with no blanks.
- **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 footing.

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
                            $12.00
1687       SOP TRUCK         $9.95 Service Office Products
                             $9.60
                             $9.90
                             $9.95
                             $9.15

           ***             $286.80


Shipping Methods
```

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 any string value enclosed in double quotes (").

options 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.

```
:SORT INVOICE BY VIA BREAK-ON VIA TOTAL EXT.PRICE COMPANY
GRAND-TOTAL "Grand Total 'LP'"

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
```

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 (') wherever 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.

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.

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).

```
:SORT INVOICE BY VIA BREAK-ON VIA "'BP'" TOTAL EXT.PRICE
COMPANY HEADING "PAGE 'PN'     Shipping Method:  'BL'"
```

sorts by shipping method, breaks after change in shipping method, creates a new page after each break, inserts break-on value in heading.

```
PAGE 1      Shipping Method:  SOP TRUCK

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
  .
  .
  .

PAGE 2      Shipping Method:  Truck

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

1681        Truck             $19.20 Kelley Brothers
                               $9.95
                               $9.95
                               $9.90
```

ID-SUPP

The ID-SUPP modifier suppresses the display of item-IDs.

If the ID-SUPP modifier is used with REFORMAT or SREFORMAT, it 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 in an Ultimate RECALL statement.

```
:SORT  INVOICE  COMPANY.L  ADDRESS  CITY  STATE  ZIP.L  ID-SUPP

PAGE   1                                              12:07:49  09 JUL 1992

Company Name..... Address..... City..... State Zip..

Kelley Brothers      12345 Main Street    Anaheim              CA    92006
Kelley Brothers      12345 Main Street    Anaheim              CA    92006
Service Office Produ 3114 Paradise Drive  San Fernando         CA    91340
Quality Lighting Pro 5454 W. Mariposa     Downey               CA    91710
Service Office Produ 4512 Orange          Santa Ana            CA    92222
Universal Copiers    211 Westgate         Long Beach           CA    91832

6 items listed.
```

LPTR

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.

NOPAGE

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.

ONLY

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.

```
:SORT  INVOICE                                    lists all default attributes.

PAGE   1                                                    12:11:51  09 JUL 1992


INVOICE : 1681
Company Name    Kelley Brothers
Invoice Date    08/23/92
Description     Diet Cola    Regular Cola    Diet Root Beer    Root Beer    Apple Jui
                ce    Tomato Juice    Crackers    Water    Raspberry Soda    Wild Cher
                ry Soda    Orange Juice    Grapefruit Juice    Herb Tea    Regular Te
                a    Cookies
Qty    2    1    1    1    2    1    1    1    1    1    4    1    1    1    6
Delivery Date    08/20/92    08/14/92    08/12/92    08/18/92    08/18/92    08/18/92
                    08/18/92    08/18/92    08/18/92    08/18/92    08/18/92    08/18
                 /92    08/18/92    08/18/92    08/18/92
Ext Price        $19.20         $9.95         $9.95         $9.90        $30.80
                $9.60        $10.30        $11.10        $10.40        $10.40
             $61.60        $12.80        $20.25         $3.05        $40.20
   .
   .
6 items listed.

: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.

```
:SORT INVOICE BY COMPANY TOTAL EXT.PRICE BREAK-ON COMPANY
DET-SUPP

PAGE   1                                          14:18:35  09 JUL 1992

INVOICE... Ext Price. Company Name.....

              $353.85 Kelley Brothers

              $220.95 Quality Lighting Products

              $286.80 Service Office Products

               $55.80 Universal Copiers
***           $917.40

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

PAGE   1                                          10:34:09  22 JUN 1992

TEST : 2
Company    Service Office Products
Contact    Lee
Address    4512 Orange
City       Santa Ana, CA 92222
Phone Number   714/555 1234

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'

PAGE   1                                              10:56:41  22 JUN 1992

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.
```

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	FOR
AN	IN
ANY	ITEMS
ARE	OF
DATA	PAGE
FILE	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)

Option	Commands that Use Option	Description
A	LIST SORT	allows forms alignment. This option is used by forms only.
B	LIST SORT	prints background item. This option is used by forms only.
C	LIST SORT	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.)
C	LIST-LABEL SORT-LABEL	suppresses the page heading line and top-of-form commands. (This is equivalent to the COL-HDR-SUPP modifier.)
C	REFORMAT SREFORMAT	if the destination of the reformatted items is tape, suppresses the tape label. (This is equivalent to the COL-HDR-SUPP modifier.)
D	LIST LIST-LABEL SORT SORT-LABEL	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.)
F	LIST-ITEM SORT-ITEM	forces a new page after every item.
H	LIST LIST-LABEL REFORMAT SORT SORT-LABEL SREFORMAT	suppresses the page heading line and the "n items listed" message at the end of the report. (This is equivalent to the HDR-SUPP or SUPP modifiers.)

Table 2-4. Options (2 of 3)

Option	Commands that Use Option	Description
H	S-DUMP T-DUMP	suppresses the tape label. (This is equivalent to the HDR-SUPP or SUPP modifiers.)
I	LIST LIST-ITEM LIST-LABEL S-DUMP SORT SORT-ITEM SORT-LABEL T-DUMP	suppresses item-ID listing. (This is equivalent to the ID-SUPP modifier.)
I	REFORMAT SREFORMAT	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.)
M	LIST SORT	lists multiple items per page. This option is used by forms only.
N	All	suppresses wait at end of page. (This is equivalent to the NOPAGE modifier.)
O	T-LOAD	overwrites the item in the file if the item on tape has same item-ID.
P	All	routes output to the spooler. (This is equivalent to the LPTR modifier.)
S	HASH-TEST ISTAT	suppresses the histogram and shows only summary statistics.
S	LIST-ITEM SORT-ITEM	suppresses display of line numbers.

Table 2-4. Options (3 of 3)

Option	Commands that Use Option	Description
W	LIST LIST-LABEL REFORMAT SELECT SORT SORT-LABEL SREFORMAT SSELECT	allows BASIC subroutine to write to files it opens.
X	LIST-ITEM SORT-ITEM	displays output in hexadecimal.
Z	LIST SORT	resets page number to 1 for each form. This option is used by forms only.

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)

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

Table 3-1. Ultimate RECALL Commands (2 of 2)

Command	Description
SREFORMAT	produces a new file from selected items and attributes, sorted by specified attribute values, on disk or tape.
SSELECT	produces a select-list that contains selected item-IDs and, optionally, specified attributes, sorted by specified attribute values.
STAT	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.
SUM	outputs the total value of a specified attribute in all selected items.
T-DUMP	outputs selected items to tape.
T-LOAD	loads selected or specified file items from the attached tape.

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.

```
:CHECK-SUM  INVOICE                    Checksums the entire file.

Byte statistics for : INVOICE
Total = 1226  Average = 204.33  Items = 6 Cksum = 14748788 Bits =
5183

:CHECK-SUM  INVOICE  NAME              Checksums the attribute NAME.

Byte statistics for : Contact
Total = 34  Average = 5.66  Items = 6  Cksum = 17601  Bits = 150
```

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 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 X'7FFFFFFF').

```
:COUNT  INVOICE

6 items counted.

:COUNT  INVOICE  WITH  NAME  LIKE  "JERRY"

2 items counted.
```

HASH-TEST

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>:
```

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.
```

ISTAT

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.

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= INVOICE 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,

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 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.

```
:LIST  INVOICE  DESC.L  QTY.L  EXT.PRICE  '1681''1682'

PAGE   1                                          17:19:11  10 JUL 1992

INVOICE... Description..... Qty. Ext Price.

1681       Diet Cola                    2      $19.20
           Regular Cola                 1       $9.95
           Diet Root Beer               1       $9.95
           Root Beer                    1       $9.90
           Apple Juice                  2      $30.80
           Tomato Juice                 1       $9.60
           Crackers                     1      $10.30
           Water                        1      $11.10
           Raspberry Soda               1      $10.40
           Wild Cherry Soda             1      $10.40
           Orange Juice                 4      $61.60
           Grapefruit Juice             1      $12.80
           Herb Tea                     1      $20.25
           Regular Tea                  1       $3.05
           Cookies                      6      $40.20
1682       Diet Cola                    2      $19.20
           Orange Juice                 1      $15.40
           Cranberry Juice              1      $12.80
           Grapefruit Juice             1      $12.80
           Hot Chocolate Regular        1       $7.35
           Hot Chocolate Sugar Free     3      $16.80

2 items listed.


:LIST  INVOICE  TOTAL  EXT.PRICE  DET-SUPP

PAGE   1                                          17:20:04  10 JUL 1992

INVOICE... Ext Price.

***           $917.40

6 items listed.
```

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.

(options 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.

Description

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                                          17:27:19  10 MAR 1992

     1682
001 Kelley Brothers
002 9003
003 7055]6032]6065]6068]5011]5015
004 2]1]1]1]1]3
005 8892]8997]8997]8997]8997]8997
006 0]0]0]0]0]0
007 Jerry
008 12345 Main Street
009 92006
010 Net 30
011 Truck
012 7145556789
```

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.)

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{,{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:

```
(count * (size + space) + indent) <= 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 V/TYP and V/MAX. 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.)

```
:LIST-LABEL  CUSTOMERS  (CI
```

produces a continuous forms format without page breaks.

```
?2,4,2,,25,6
Quality Lighting Products       Service Office Products
Mat                             Lee
5454 W. Mariposa                4512 Orange
Downey, CA 91710                Santa Ana, CA 92222


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
```

REFORMAT

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

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 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**.

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.

```
:REFORMAT INVOICE COMPANY NAME ADDRESS CITY STATE ZIP *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.

```
:REFORMAT TEMP *A9998 *A1 *A0 *A2 *A3 *A4 *A5 *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.

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

attaches the tape at the size of the records that will be written by REFORMAT.

```
:REFORMAT CUSTOMERS COMPANY NAME ADDRESS CITY STATE ZIP
(H

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

creates a file on tape with company as the first value, followed by name, address, city, state, and zip. The tape label is suppressed. The length of the each tape block is 100 characters.

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

rewinds the tape to display the tape records.

```
:T-READ
Block size: 100

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

Record = 2

   1 Service Office Products       Lee
  51 4512 Orange         Santa Ana           CA   92222
   .
   .
End of file
```

displays the tape records.

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.)

(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.

```
: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

L 2000#11:56:41  12 MAR 1992 CUSTOMERS S-DUMP Format                    ~01

Record = 1

   1 1^Quality Lighting Products^Mat^5454 W. Mariposa^D
  51 owney^CA^91710^3015551234^[2^Service Office Produc
 101 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^Servi
 301 ce Office Products^Sheiby^3114 Paradise Drive^San
 351 Fernando^CA^91340^8185551234^[_X{{{{{{{{{{{{{{{{{
 401 {{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{
 451 {{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{
 501 {{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{
  .
  .
8051 {{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{
8101 {{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{
8151 {{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{


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 = "O"

4 items selected.

:SORT INVOICE COMPANY TOT.PRICE

PAGE 1                                          11:23:26  12 MAR 1992

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


PAGE   1                                        09:44:59  12 MAR 1992

INVOICE... Company Name.................. Invoice.. Tot Price.
                                          Date

1696       Universal Copiers             07/01/92      $55.80
1686       Quality Lighting Products     08/21/92     $220.95
1681       Kelley Brothers               08/23/92     $269.50
1682       Kelley Brothers               08/24/92      $84.35

***                                                    $630.60

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


PAGE   1                                        09:55:51  12 MAR 1992

INVOICE... Product... Description.............. Ext Price.
           Number

           7001       Water                          $11.10
           7015       Raspberry Soda                 $10.40
           7017       Wild Cherry Soda               $10.40
           7055       Diet Cola                      $57.60
           7056       Regular Cola                   $29.85
           7065       Diet Root Beer                 $19.90
           7066       Root Beer                      $19.80
           7331       Tomato Juice                   $19.20
           7345       Apple Juice                    $77.00
           8036       Cookies                        $40.20
           8123       Crackers                       $20.60
***                                                 $316.05

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.)

	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.

(options 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.

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 INVOICE WITH ZIP = "917]"

PAGE   1                                          10:00:35  12 MAR 1992

    1686
001 Quality Lighting Products
002 9000
003 7056]7055]7066]7065]2025]6024]6032]0005]3007]3035]3004]5003]8123
004 2]2]1]1]4]1]2]1]1]1]1]1]1
005 8989]8989]8989]8989]8989]8989]8989]8989]8989]8989]8989]8989]8989
006 O
007 Mat
008 5454 W. Mariposa
009 91710
010 Net 10
011 Truck
012 3015551234

:
```

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:

```
(count * (size + space) + indent) <= 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 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.

```
:SORT-LABEL CUSTOMERS BY ZIP (CI
?2,4,2,13,24,6
?COMPANY
?CONTACT
?STREET
?CITY, STATE

COMPANY       Service Office Products       Quality Lighting Products
CONTACT       Shelby                        Mat
STREET        3114 Paradise Drive           5454 W. Mariposa
CITY, STATE   San Fernando, CA 91340        Downey, CA 91710


COMPANY       Universal Copiers             Kelley Brothers
CONTACT       Marina                        Jerry
STREET        211 Westgate                  12345 Main Street
CITY, STATE   Long Beach, CA 91832          Anaheim, CA 92006


COMPANY       Service Office Products
CONTACT       Lee
STREET        4512 Orange
CITY, STATE   Santa Ana, CA 92222
```

SREFORMAT

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

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.

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.

SSELECT

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.

```
:SSELECT  INVOICE  BY-EXP  PROD.NO          Select items in order by
                                             product number.

52 items selected.

:SAVE-LIST  PROD.NO                          Saves list for future use.

'PROD.NO' saved - 1 frames used.

:GET-LIST  PROD.NO                           Gets previously saved list.

52 items selected.

:LIST  INVOICE  PROD.NO  DESC                The list is in sorted order by
                                             product number.

PAGE  1                                              11:03:46  30 JUN 1992

INVOICE... Product... Description....
           Number

1681       0005       Herb Tea
1683       0005       Herb Tea
1686       0005       Herb Tea
1681       2025       Regular Tea
1683       2025       Regular Tea
1686       2025       Regular Tea
1687       2025       Regular Tea
  .
  .
52 items listed.

:
```

STAT

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 =    $636.35 Average =    $106.0583 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:

```
Total of name = t
```

where

name attribute name, if specified; otherwise filename.

t 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 =    $636.35
```

T-DUMP

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.)

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.

```
:T-ATT

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

L 2000#15:28:07  12 MAR 1992 INVOICE                                    ~01

Record = 1
   1 1686^Quality Lighting Products^9000^7056]7055]7066
  51 ]7065]2025]6024]6032]0005]3007]3035]3004]5003]8123
 101 ^2]2]1]1]4]1]2]1]1]1_[]1]1]1^8989]8989]8989]8989]8
 151 989]8989]8989]8989]8989]8989]8989]8989]8989^O]O]O]
 201 O]O]O]O]O]O]O]O]O^Mat^5454 W. Mariposa^917_[10^Net
 251  10^Truck^3015551234^[1683^Service Office Products
 301 ^8988^7345]7331]8123]6032]6068]0005]2025]8036]4241
 351 ]5003]5005^3]1]3]1]1]1]2]4]1]1]1^8892]8892_[]8892]
  .
  .
8051 [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
8101 [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[
8151 [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[


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.
```

Notes

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

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

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
```

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

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

D/CODE - Definition Code

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.

A/AMC - Attribute Number

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.

S/NAME - Column Heading

The S/NAME attribute (attribute 3) is an optional field. If used, it contains the column heading to use for this attribute on 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

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 **amc*n*** is the attribute number or name of the nth dependent attribute.

A dependent attribute has the following structure code format:

```
D;amc
```

where **amc** is the controlling attribute number or name.

```
     PROD.NO                    DESC
001  A                     001  A
002  3                     002  991
003  Product]Number        003  Description
004  C;DESC;PRICE;QTY      004  D;PROD.NO
      DEL.DATE                   .
 .                               .
 .

     PRICE                      QTY
001  A                     001  A
002  992                   002  4
003  Price                 003  Qty
004  D;PROD.NO             004  D;PROD.NO
 .                               .
 .                               .
```

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).

```
     COMPANY.L
001  A
002  1
003  Company Name
004
005
006
007  MCT]ML(#20)
008
009  L
010  20
```

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.

Table 4-3. V/TYP Codes

Code	Placement of Text	Sorting Considerations
L	left justifies data. If the number of characters in the attribute exceeds the column width, the extra characters are wrapped to the next line.	alphanumeric (ASCII) sort.
R	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.	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).
RN	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.	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.
T	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.	alphanumeric (ASCII) sort.
U	left justifies data. Ignores V/MAX; text continues to end of attribute, wraps at end of screen. Used primarily for last attribute on line.	alphanumeric (ASCII) sort.

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 alpha-numerically 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.

item-ID	SORT.LEFT	SORT.RIGHT	SORT.RIGHTN
001	A	A	A
002	1	1	1
003			
004			
005			
006			
007			
008			
009	L	R	RN
010	10	10	10

Figure 4-1. Multiple Attribute Definition Items

Data	SORT.LEFT	SORT.RIGHT	SORT.RIGHTN
-123	-123	-123	100A
123.12	-123.12	-123.12	10B
-123.12	100A	1C	1C
20	10B	2	C1
2	123	10B	-123.12
10B	123.12	20	-123
C1	1C	100A	2
100A	2	123	20
1C	20	123.12	123
123	C1	C1	123.12

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

Notes

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)

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

Table 5-1. Processing Codes (2 of 2)

Code	Type	Description
Tfile	file translation	verifies or converts attribute values by translating them through a specified file.
U	user-defined	invokes user-written routines.
V	item-ID extraction	specifies attribute that contains item-IDs for WITHIN connective.

Correlatives and Conversions

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

Format Type	Description
internal	format of data when retrieved from storage; the stored format.
intermediate	format created by Ultimate RECALL for processing; the processing format.
external	format created by Ultimate RECALL for printing or displaying; the output format.

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

definition item has a conversion code, and the Ultimate RECALL statement specifies a value for that attribute, the value in the Ultimate RECALL statement is converted to the processing format before using it. If the attribute has both a correlative and a conversion, both stored values and input values are converted before comparing.

Multiple Processing Codes

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]MR2$,(#10)
```

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:

```
[172] Self-referential A-correlative error.
      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.

`1*2+3<4`	is equivalent to	`((1*2)+3)<4`
`4>=5-2/1`	is equivalent to	`4>=(5-(2/1))`

```
1+2-3                    is equivalent to      (1+2)-3

4/5*6                    is equivalent to      (4/5)*6

A IF 6['1','1'] = '*' THEN IF 2 > "10" THEN 'OK' ELSE
'' ELSE ''
                         is equivalent to:
          A IF 6['1','1'] = '*' THEN (IF 2 > "10" THEN
          'OK' ELSE '') ELSE ''
```

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.

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.

```
:CT  DICT  INVOICE  AVERAGE
    Average
001 A
002 0
003 Average Owed
004
005
006
007 A;LPV/ND*MR2S(#1,2)
008 A4*3(TPRCD.NO;X;;;;)
009 R
010 12

:

:SORT  INVOICE  TOTAL  AVERAGE  DET-SUPP

PAGE    1                                          16:10:09  16 JUN 1992

INVOICE... Average Owed

***                1  2.33

6 Items listed.
```

Processing Code	Data in Attribute	Results
A1+2	001 213 002 203	416
A"10"*3	003 96	960
AS(5+"100")	005 1]2]3	306
A;(NI)/"43"	If NI = 42 If NI = 44	0 1
A;2R * 3	002 2 003 1]2]3	2]4]6
A;2 * 3	002 2 003 1]2]3	2]0]0
A;IF 1 > "100" AND 2 < 3 THEN "EUREKA" ELSE ""	001 101 002 90 003 95	EUREKA
A IF 9(L0) = '9' THEN 9(ML#5-#4) ELSE 9(ML#5)	009 92006 009 920061234	92006 92006-1234
A;IF 1 > '100' THEN '*REORDER*' ELSE ''	001 254	*REORDER*
A1['1','999'*(2=3)]	001 12334567890 002 350 003 350 001 12334567890 002 350 003 351	1234567890 (null)
A(N(PRICE)*N(QTY))(MR2$) (where PRICE is attribute 5 and QTY is attribute 6)	005 2550]3500 006 1000]1000	$25500.00 $35000.00

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

;	semicolon; required separator.
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

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

Table 5-3. The C.FLAGS Variable

Value	Meaning
0	processing a detail line.
1-15	processing a BREAK line (subtotal line); the C.FLAGS value is the break level number.
127	processing GRAND-TOTAL line.
-1	processing a sort or selection.

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 1
 .
 .
007 ML(###/### ####)
008 B;TRANS.PHONES
009 L
010 15
```

Attribute definition item for the CONTACT# attribute:

```
001 A
002 1
 .
 .
009 L
010 10
```

Subroutine TRANS.PHONES:

```
001 SUBROUTINE TRANS.PHONES
002 $INCLUDE SYSLIB RCL.COMMON
003 IF FIRST ELSE
004  FIRST = 1      ;* open file only once in report
005  OPEN 'CONTACTS' [illegible] 'CONTACTS'
006 END
007 READV C.DATA [illegible]
008 RETURN
```

The READV 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

PAGE   1                                   [illegible] JAN 1992

CONTACTS.X [illegible] ..........

ULT      [illegible]

           [illegible]

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 \
 .
 .
007 B;RUNNING.TOTAL MR2S,(#10)
008 A4*3(IPROD.NO;X;2;2)
009 L
010 0
```

Subroutine RUNNING.TOTAL:

```
001 SUBROUTINE RUNNING.TOTAL
002 $INCLUDE SYSLIB RCL.COMMON
003 IF C.FLAGS > 0 THEN
004 * break [illegible]
005  C.DATA = TOTAL
006 END ELSE
00  * normal [illegible]
008  IF NUM(C.DATA) [illegible]
009 * safety [illegible]
010 END
011 RETURN
```

Ultimate RECALL displays the running total as follows:

```
:SORT INVOICE BY ZIP BREAK-ON ZIP.L "'L'" TOTAL TOT.PRICE
CUMULATIVE
PAGE [illegible]                                    [illegible] JAN 1992

INVOICE...  [illegible]

1683        [illegible]
            ***         [illegible]

1686        [illegible]
            ***         [illegible]

1696        [illegible]
            ***         [illegible]

1688        [illegible]
1691        [illegible]
            ***         [illegible]

188         [illegible]
            ***         [illegible]

***                     [illegible]
6 items listed.
```

C - Concatenation Code

The C code concatenates attributes and literal values.

Syntax

C{;}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.

Code	Attribute Value	Results
C2;' ANS'=3+1	001 42 002 THE 003 1000	THE ANS=1000+42
C"NAME":4,5	004 STERN 005 FRANK	NAME:STERN,FRANK
C*%6	006 INTEREST last computed value: 12.73	12.73 INTEREST
C9;10	009 PT# 010 203	PT#203

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 in the date. 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**.

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:

Internal (Stored) Format	External (Output) Format
-100	22 SEP 1967
-10	21 DEC 1967
0	31 DEC 1967
1	01 JAN 1967
100	09 APR 1968
1000	25 SEP 1970
10000	18 MAY 1995

Code	Attribute Value	Result
D	8888	01 MAY 1992
D2	8888	01 MAY 92
D4	8888	01 MAY 1992
D0	8888	01 MAY
D-	8888	05-01-1992
D2/	8888	05/01/92
D2*	8888	05*01*92
D/	8888	05/01/1992
D"1	INV"8888	INV"01 MAY 1992
DY	8888	1992
D2Y	8888	92
DQ	8888	2 (second quarter)
DD	8888	1 (first day)
DM	8888	5 (fifth month)
DJ	8888	122 (Julian date)

F - Function Code

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 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 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 stack1 and stores the sum in stack1.

Caution: *Although the S operator sums only the values in stack1, 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 stack1 and stack2.

P — pushes the value in stack1 back onto the stack; that is it duplicates the stack 1 value and stack1 and stack2 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 stack1 value; the converted value replaces the original value in stack1. The code can include parentheses, as in pattern matching and output masks.

relational operators

tests stack1 and stack2 values to determine if the stated relation is true or false. If true, a 1 (one) is stored in stack1; if false, a 0 (zero) is stored in stack1.

= — equal-to; tests if stack1 value is equal to stack2 value.

< — less-than; tests if stack1 value is less than stack2 value.

> — greater-than; tests if stack1 value is greater than stack2 value.

— not-equal; tests if stack1 value is not equal to stack2 value.

[— less-than or equal-to; tests if stack1 value is less than or equal to stack2 value.

] — equal-to or greater-than; tests if stack1 value is equal or greater than stack2 value.

Description

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;/*
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 stack1. 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 stack1. The values are processed and the result pushed on to stack1. This push-down, pop-up process repeats until the entire F code has been processed. Then the stack1 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;+;*
```

	C3; push constant 3 onto stack	C4; push constant 4 onto stack	C20; push constant 20 onto stack	+; add stack1 and stack2	* multiply stack2 by stack1
stack1	3	4	20	24	72
stack2		3	4	3	
stack3			3		

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:

	2;	3;	*;	C#;	:	1;	C3;	C2;	[];	:
stack1	20	40	800	#	800#	W/29	3	2	29	800#29
stack2		20		800		800#	W/29	3	800#	
stack3							800#	W/29		
stack4								800#		
stack5										

Figure 5-2. F-Code Stack Activity to Resolve F;2;3;*;C#;:;1;C3;C2;[];:

Totals

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.

```
:CT  DICT  INVOICE  F.AVERAGE

    F.AVERA
001 A
002 S
003 Average
004
005
006
007
008
009 R
010

:

:SORT  INVOICE  TOTAL  F.AVERAGE  DET-SUPP

PAGE                                                    JUL 1992

INVOICE...

***

6
```

General Examples

The following examples illustrate general uses for F processing codes.

Code	Attribute Value	Results
F;1;2;*	001 20 002 40	800
F;1;2;*	001 20 30 002 40	800 0
F;1;2R;*	001 20 30 002 40	800 1200
F;1;S;2;*	001 20 30 002 40	2000
F;1;3;(TPROD.NO;X;2;2);*	001 20 002 40 003 7015	20800
PROD.NO file, item 7015	002 1040	
F;1;(ML#10);2;:	001 20 002 40	20 40
F;D;(DY);4;(DY);-	004 7900	3 (if current year is 1992)

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.

***** 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.

Code	Attribute Value	Results
G/1	05/01/92	05
G1/1	05/01/92	01
G2 1	01 MAY 1992	1992
G 2	01 MAY 1992	01 MAY
G1 2	01 MAY 1992	MAY 1992
G-1	123-455-789	123
G5-1	123-455-789	(null)
G1*1	*START HERE	START HERE
G1$2	123$456$789	456$789
G#3	40#50#60#70#80	40#50#60#70#80#

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

Ln{,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.

Code	Attribute Value	Results
L5	92705	92705
L9	92705	(null)
L5,9	92705	92705
L6,8	92705	(null)
L0	92705	5

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.

Code	Attribute Value	Results
MCA	Alpha and 1234	Alphaand
MC/A	Alpha and 1234	1234 (two spaces precede the numbers)
MCN	Alpha and 1234	1234
MC/N	Alpha and 1234	Alpha and
MCL	Alpha and 1234	alpha and 1234
MCU	Alpha and 1234	ALPHA AND 1234
MCT	Alpha and 1234	Alpha And 1234
MCD	1234	4D2
MCX	1234	4660
MCP	X'0041FB' (3 bytes)	.A.

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 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:

#n	left-justifies value in a field of n blanks.
***n**	left-justifies value in a field of n asterisks.
%n	left-justifies value in a field of n zeros.
&n	left-justifies value in a field of n blanks.
x{n}	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.

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).

Code	Attribute Value	Results
ML0	4999	4999
ML2	4999	49.99
ML02	4999	50
ML20,	4999	4,999.00
ML2E	-123400	<1234.00>
ML2M	-123400	1234.00-
ML2,M	-123400	1,234.00-
ML4	987654	98.7654
ML2Z	001234	12.34
ML(%10)	55	5500000000
ML(*5)	55	55***
ML,$(#9)	123456	$123,456
ML,($#9)	123456	$123,456
ML,$(*20)	123456789	$123,456,789********
ML(#2^3#2^#4)	123456789	12^^^34^5678
ML(%%%-%%-%%%%)	123456789	123-45-6789
ML(#3-#4 EXT.#2)	123456789	123-4567 EXT.89

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.

Code	Attribute Value	Results
MP	X'342F'	342
MP	X'343D'	-343
MP	X'01100F'	1100
MP	X'3D'	-3
MP	X'010D'	-10
MP	X'1F'	1
MP	X'723F'	723
MP	X'099F'	99
MP	X'78764D'	-78764

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:

#n	right-justifies value in a field of n blanks.
***n**	right-justifies value in a field of n asterisks.
%n	right-justifies value in a field of n zeros.
&n	right-justifies value in a field of n blanks.
x{n}	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.

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 (nnnnnnnnn) 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.

Code	Attribute Value	Results
MR0	4999	4999
MR2	4999	49.99
MR02	4999	50
MR20,	4999	4,999.00
MR2E	-123400	<1234.00>
MR2M	-123400	1234.00-
MR2,M	-123400	1,234.00-
MR4	987654	98.7654
MR2Z	001234	12.34
MR(%10)	55	0000000055
MR(*5)	55	***55
MR,$(#9)	123456	$123,456
MR,($#9)	123456	$ 123,456
MR,$(*20)	123456789	********$123,456
MR(#2^3#2^#4)	123456789	23^^^45^6789
MR(%%%-%%-%%%%)	123456789	123-45-6789
MR(#3-#4 EXT.#2)	123456789	123-4567 EXT.89

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

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 Value	Results
MT	43203	12:00
MTH	43203	12:00PM
MTS	43203	12:00:03
MTHS	43203	12:00:03PM
MT	(null)	(blank)

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.)

Code	Attribute Value	Results
MX	(4 spaces)	20202020
MX	+1	2B31
MX	<ESC>A	1B41

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

P(pattern){;(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.

Code	Attribute Value	Results
P(2N)	12	12
P(3N4A)	123 Main	(null)
P(3N 4A)	123 Main	123 Main
P(0N 4A)	123 Main	123 Main
P(ULT-3A)	ULT-CAL	ULT-CAL
P(2X*4X)	67*AB21	67*AB21
P(2X*4X)	ZZ*1224	ZZ*1234
P(2X*4X)	00*1234	00*1234
P(2X*4X)	00*12345	(null)
P(1A);(2N)	D	D
P(1A);(2N)	34	34
P(1A);(2N)	T1	(null)
P(2N$2A-3X)	12$AB-(*)	12$AB-(*)

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

Rn;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.

Code	Attribute Value	Results
R1;5	5	5
R1;5	927	(null)
R-10;-1	-3	-3
R1;100	100A	(null)
R1;4,6;9	6	6
R1;4,6;9	5	(null)

S - Substitution Code

The S code substitutes an attribute value with a different value.

Syntax

S;non-null;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.

Code	Attribute Value	Results
S;'AFFIRM';'N/A'	Y	AFFIRM
S;'AFFIRM';'N/A'	(null)	N/A
S;'AFFIRM';'N/A'	Y]Y]]Y	AFFIRM]AFFIRM]N/A] AFFIRM
S;1;2 (where 3 is current attribute)	001 *]*]* 002 -]-]- 003 Y	* * *
S;1;2 (where 3 is current attribute)	001 *]*]* 002 -]-]- 003 (null)	- - -
S;*;'NO BAL'	00	NO BAL
S;*;'NO BAL'	100	100

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.

Code	Justification	Attribute Value	Results
T2	L	30#650	30
T2	R	30#650	50
T4,2	L	30#650	65
T4,2	R	30#650	65
T8,2	L	30#650	(null)

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 Tfile code is used as a conversion (attribute 7).

Description

A Tfile 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 Tfile 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 Tfile 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-amc 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:

```
item-ID     92714          92705
    001     Irvine
    002     CA             CA
```

When the following data is used, the results for output are as shown (note that 92716 is not in the ZIP file):

Code	Attribute Value	Result
TZIP;C;1;1	92714	Irvine
TZIP;C;1;1	92716	92716
TZIP;I;1;1	92714	Irvine
TZIP;I;1;1	92716	92716
TZIP;O;1;1	92714	Irvine
TZIP;O;1;1	92716	[708] '92716' cannot be converted.
TZIP;V;1;1	92714	92714
TZIP;V;1;1	92716	[708] '92716' cannot be converted.
TZIP;V;1;1	92705	[708] '92705' cannot be converted.
TZIP;X;1;1	92714	Irvine
TZIP;X;1;1	92716	(null)

```
:CT  DICT  INVOICE  VERIFY.ZIP

    VERIFY.ZIP
001 A
002 9
003
004
005
006
007
008 A9["1","5"](TZIP;V;1;1)
009 L
010 20

:
:SELECT  INVOICE  WITH  VERIFY.ZIP

6 items selected.

:SORT  INVOICE  COMPANY  CITY  STATE  ZIP
INVOICE... Company [illegible]........ [illegible]................   [illegible]..

1681        Kelley Brothers      Anaheim              CA    [illegible]2006-1234
1682        Kelley Brothers      Anaheim              CA    [illegible]2006
1683        Service Office [illegible]                CA    [illegible]
1686        Quality [illegible] Downey                CA    [illegible]
1687        Service Office [illegible]Santa Ana       CA    [illegible]2222
1696        Universal Coolers    Long Beach           CA    [illegible]1832

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

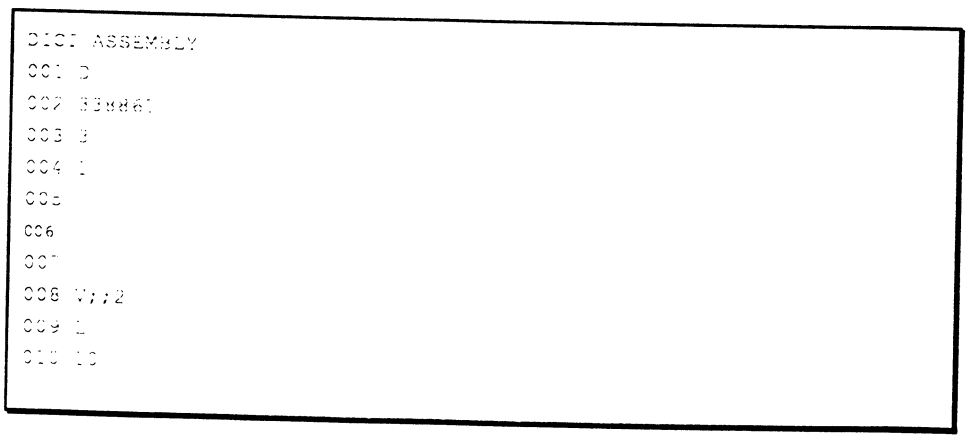
;;	two semicolons; required separators.
amc	number of attribute that contains the item-IDs.

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.



Notes

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.*

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

Company Name	Address	City	State	Zip
Kelley Brothers	12345 Main Street	Anaheim	CA	92006

Contact	Date	Terms	Ship Via
Jerry	08/24/92	Net 30	Truck

Quantity	Product Number	Description	Unit Price	Total Price	Delivery Date
2	7055	Diet Cola	$9.60	$19.20	05/05/92
1	6032	Orange Juice	$15.40	$15.40	08/18/92
1	6065	Cranberry Juice	$12.80	$12.80	08/18/92
1	6068	Grapefruit Juice	$12.80	$12.80	08/18/92
1	5011	Hot Chocolate Regular	$7.35	$7.35	08/18/92
3	5015	Hot Chocolate Sugar Free	$5.60	$16.80	08/18/92
			Total	$84.35	

Figure 6-1. Sample Forms Report

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.

- 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.

Forms Expression

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

```
@{p}(c,r{,"message"}):name{[1,z]}
@{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.

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  0
     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#
```

`@(15,7):NAME`	prints the value of NAME starting at column 15 on row 7 of every page of the form set.
`@A(5,15):DESC`	prints the value of DESC starting at column 5 on row 15 of every page of the form set.
`@M(1,12,"** MORE **"):N`	prints the message ** MORE ** on all but the last page of each form. Prints the value of N on last page.

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:

```
@C(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]} ...{@(cn{,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.
c*n*	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.
y*n*	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.

attrib-name*n*	attribute name of the nth attribute to be displayed.
[1,z*n*]	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:

```
Column: 01234567890123456789O
    R  0
    o  1
    w  2
       3
       4      7055       Diet Cola              $19.20
       5      7056       Regular Cola            $9.95
       6      7065       Diet Root Beer          $9.95
       7      7066       Root Beer               $9.90
       8      7345       Apple Juice            $30.80
       9
```

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:

```
Column: 012345678901234567890123456789012345678901234567890
    R  0
    o  1
    w  2
       3
       4      7055       Diet Cola
       5                                         $19.20
       6      7056       Regular Cola
                                                 $9.95
       7      7065       Diet Root Beer
       8                                         $9.95
```

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
- HEADING
- ID-SUPP
- ONLY
- HDR-SUPP
- 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.

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.

```
 TEMPLATE:
001
002    ****************************************
003    * * * * * * * * * * * * * * * * * * * *
004    ****************************************
005
006    INVENTORY
007    LISTING
008    ========
```

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).

```
:SORT INVOICE @(3,2):NAME @(3,3):COMPANY @(3,4):ADDRESS
@(3,5):CITY.L @(23,5):STATE @(27,5):ZIP.L (HM

Subpage size>5

   Jerry
   Kelley Brothers
   12345 Main Street
   Ananeim              CA  92006

   Jerry
   Kelley Brothers
   12345 Main Street
   Ananeim              CA  92006

   Shelby
   Service Office Products
   3114 Paradise Drive
   San Fernando         CA  91340

   Mat
   Quality Lighting Products
   5454 W. Mariposa
   Downey               CA  90210

   Lee
   Service Office Products
   4512 Orange
   Santa Ana            CA  92707

   Marina
   Universal Software
   211 Aviation
   Long Beach           CA  90815
```

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.

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

Condition	V/TYP	On Standard Reports	On Forms
data <= V/MAX	L, T, U	prints on one row, left-justified.	prints on one row, left-justified.
	R	prints on one row, right-justified.	prints on one row, right-justified.
data > V/MAX	L	folds to new row at V/MAX character.	truncates at V/MAX character.
	T	folds to new row at word boundary prior to V/MAX.	if in window, folds to new row at word boundary; if not in window, truncates at V/MAX character.
	U	overflows to right.	overflows to right.
	R	overflows to left.	overflows to left.
V/MAX = 0	all	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.	no length test; all data is suppressed on detail lines, but printed on break rows, if specified.

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):" "
```

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*.)

INVOICE	D/CODE	AMC	S/NAME	CONVERSION	CORRELATIVE	TP	MAX
ADDRESS	A	8	Address			L	20
CITY	A	9	City	A9["1","5"](TZIP;X;1;1)		L	20
COMPANY	A	1	Company Name	MCT		U	20
DATE	A	2	Invoice,Date	D2/		L	9
DEL.DATE	A	5	Delivery,Date	D2/		L	9
DESC.L	A	991	Description	MCT ML(#25)	[illegible]	L	25
EXT.PRICE	A	91	Ext Price	[illegible]	[illegible]	R	10
INVOICE#	A	0	Invoice Number			[illegible]	10
NAME	A	7	Contact			L	20
PRICE.L	A	3	Price	MR2S[illegible]	[illegible]	R	10
PROD.NO	A	3	Product Number			L	10
QTY.L	A	4	Qty			R	4
STATE	A	9	State	A9["1","5"](TZIP;X;2;2)		L	2
TERM	A	10	Terms			[illegible]	12
TOT.PRICE	A	90	Tot Price	[illegible]	[illegible]	R	[illegible]
VIA	A	11	Ship Method			[illegible]	12
ZIP	A	9	Zip	[illegible]		[illegible]	5

17 items listed.

Figure 6-2. DICT INVOICE Items

```
    PRINT.INVOICE
001 PQ
002 * Print out invoices
003 HSORT INVOICE
004 H @(15,1):INVOICE#
005 H @(2,8):COMPANY
006 H @F(24,8):ADDRESS
007 H @F(46,8):CITY
008 H @F(68,8):STATE
009 H @F(72,8):ZIP.L
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):QTY.
016 H @(9):PROD.NO
017 H @(20):DESC.
018 H @(47):PRICE.
019 H @(58):EXT.PRICE
020 H @(70):6
021 H END-WINDOW
022 H @M(2,42,"Continued . ."):MESSAGE
023 H TOTAL @L(58,47):TOT.PRICE
024 H HDR-SUPP
025 IF A2 A2
026 H (PA
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.

```
          1681



Kelley Brothers            12345 Main Street        Anaheim          CA       92006


Jerry                      08/23/92                 Net 30           Truck
```

2	7055	Diet Cola	$9.60	$19.20	08/20/92
1	7056	Regular Cola	[illegible]	[illegible]	[illegible]
1	7065	Diet Root Beer	[illegible]	[illegible]	[illegible]
1	7066	Root Beer	[illegible]	[illegible]	08/18/92
2	7345	Apple Juice	[illegible]	[illegible]	08/18/92
1	7331	Tomato Juice	$9.60	$9.60	08/18/92
1	8123	Crackers	$10.30	$10.30	08/18/92
1	7001	Water	[illegible]	[illegible]	[illegible]
1	7015	Raspberry Soda	[illegible]	[illegible]	[illegible]
1	7017	[illegible]	[illegible]	[illegible]	[illegible]

Continued ..

Figure 6-4. Sample Multipaged Form (1 of 2)

1681

Kelley Brothers

Jerry 08/23/92 NET 30 Truck

4	6032	Orange Juice	$15.40	$61.60	08/18/92
1	6068	Grapefruit Juice	$12.80	$12.80	08/18/92
1	0005	Herb Tea	$20.25	$20.25	08/18/92
1	2025	Regular Tea	$3.05	$3.05	08/18/92
6	8036	Cookies	$6.70	$40.20	08/18/92

$269.50

Figure 6-4. Sample Multipaged Form (2 of 2)

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

Sequence	Condition	Prompt	Notes
1	B option	`Background File & Item>`	
2	M option	`Subpage size>`	
3	C code	`Audit File>` `Starting number>`	Starting number can be stacked, but is normally entered by the operator.
4	A option	`Align? Y=CR/N>`	Response cannot be stacked

7 Hints

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.

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.

Printing Dependent Attributes

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.

```
    6                          non-dependent attribute definition item.
001 A
002 5
003 Delivery Date
004
005
006
007 D2/
008
009 L
010 9

    DEL.DAT                    dependent attribute definition item.
001 A
002 5
003 Delivery Date
004 D;PROD.NO
005
006
007 D2/
008
009 L
010 9
```

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.

```
: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        08/10/92
  .
  .
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                                          12:12:04  09 JUL 1992

INVOICE... Product...  Delivery.
           Number      Date

1681                   08/18/92
1683                   05/05/92
  .
  .
1681       7056        08/14/92
1686       7056        08/10/92
1687       7056        05/16/92
1681       7065        08/12/92
1686       7065        08/10/92
  .
  .
52 items listed.
:
```

```
:SORT  INVOICE  DEL.DATE
```

the delivery date is not displayed if the product number is not specified.

```
PAGE   1                                        12:12:09  09 JUL 1992

INVOICE...


1681
1682
1683
1686
1687
1696

6 items listed.


:SORT  INVOICE  6
```

use 6 to display the delivery date even if the product number is not specified.

```
PAGE   1                                        12:12:13  09 JUL 1992

INVOICE... Delivery.
           Date

1681       08/20/92
           08/14/92
           08/12/92
           08/18/92
           08/18/92
  .
  .

6 items listed.
```

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
001 A
002 34
003 \
004
005
006
007 MR2D;#12)
008 A4*$:IFRCD.NO;A;;;;
009 L
010 0

:SORT  INVOICE  BY  DATE  DATE  TOTAL  NO-DET

PAGE    1                                          12:01:44   19 JUL 1992

INVOICE...  .*VN..DD..
            DATE

1687        MR 10 92
1696           10 92
1683        MR 10 92
1686        MR    92
1681        MR 9  92
1682        MR 24 92

***                              3911.40

6 items listed.
```

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.

```
    TEN
001 A
002 34
003 \
004
005
006
007 C;""
008 A;(NI)/"10"
009 L
010 0

:SORT  INVOICE  BY-EXP  PROD.NO  PROD.NO  TOTAL  EXT.PRICE
BREAK-ON  TEN  "  "  ID-SUPP

PAGE    1                                          16:1 :0  09 JUL 1992

Product...  Ext Price.
Number

0005              $20.25
0005              $20.25
0005              $20.25
2025               $8.0
2025               $6.
2025              $10.
2025               $4.
3004               $9.
300                $ .
3035               $1.80

                 $146.2

  .
  .
8123              $30.00
8123              $10.00

                  $41.

                  $4 .

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.

```
:LIST  INVOICE  TOTAL  EXT.PRICE  DET-SUPP

PAGE   1                                          14:10:05  09 JUL 1992

INVOICE...  EXT.PRICE.

***            $917.40
6 items listed.


:SORT  INVOICE  BY  COMPANY  BREAK-ON  COMPANY  TOTAL  EXT.PRICE
DET-SUPP

PAGE   1                                          14:10:05  09 JUL 1992

INVOICE... Company Name.....  Ext. Price.

           Kelley Brothers          $353.85

           Quality Lighting Prod    $220.95

           Service Office Produc    $286.80

           Universal Copiers        $555.80
***                                 $917.40

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.

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.

Verifying Data

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.

Notes

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.

attribute mark — delimiter used to separate attributes; has decimal ASCII value of 254.

audit file — file used by forms processing C print specification code to store audit trail information.

base, base FID — file location; first frame ID (FID) of primary storage.

BASIC — a programming language, name is acronym for Beginner's All-purpose Symbolic Instruction Code.

BASIC subroutine — 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.

block (tape) — unit of information on a tape.

braces — punctuation marks { } in syntax definitions that surround optional parameters.

checksum — 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.

columnar format — report format where the headings and data are displayed across the top of the page. *See also non-columnar format.*

complex itemlist — an itemlist that contains relational operators.

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 correlatives 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.

default value	preset value for a parameter if no user input is provided.
default output specifications	attribute definition items with item-IDs that are consecutively numbered from 1 to nn (the highest consecutive number).
delimiter	separator between elements. In an Ultimate RECALL sentence, spaces are used as delimiters.
dependent attribute	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.
dictionary	file section that contains attribute definition items and pointers to data files.
dummy AMC	an AMC that is greater than the highest actual AMC on the file.
dummy attribute	an attribute with a dummy AMC.
enter	by convention, this means type input, then press the RETURN key.
ERRMSG file	file containing system messages.
error message	system response to incorrect entry or other processing error.
explicit itemlist	the item-IDs of the items to be included in the report are explicitly named in the Ultimate RECALL statement.
extended items	items with over 32Kb characters.
FID	frame ID or frame number.
file	contains one or more items.
file definition item	defines and points to a file in the current account. Also known as a D-pointer.

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.

intermediate format	format created by Ultimate RECALL for processing; correlatives are applied to the stored data to change it to the intermediate format.
internal format	format in which data is stored.
item	a set of related attributes; also known as record.
item-ID	name by which an item is identified. Also known as key record.
itemlist	part of an Ultimate RECALL statement that specifies items to be considered.
justification	the alignment of data in an item for display or sorting.
line	data line between the terminal and the system.
logoff	ends a work session at a terminal and make it inactive.
logon	starts a work session at a terminal in a specified account.
logto	exits the current account and starts a session in the specified account.
M/DICT	heading for the Master Dictionary.
master dictionary	contains the account's verbs, PROCs, and cataloged BASIC programs, as well as D-pointers and Q-pointers.
MD	master dictionary.
message line	last line of report; shows the number of items on the report
modifier	connective that controls the format and layout of a report, such as double-spacing.

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.

print limiter — indicates criteria that a value in the specified attribute must meet in order to be included on the report.

PROC — executable program written in Ultimate PROC procedural language. Can be used as an Ultimate command.

processing code — 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.

processing format — another name for intermediate format; called processing because both stored and user-supplied values are converted to it before most processing takes place.

Q-pointer — see *file synonym definition item.*

record (tape) — see *block.*

relational operator — connective used to test relationships between elements. For example, = or NOT are relational operators.

report body — contains the detail lines of the report as specified in the statement.

RETURN — the keyboard RETURN or ENTER key.

reverse Polish notation — method of specifying a function or formula where operands are specified before the operator; reverse Polish notation is used in F processing codes.

S/AMC — report heading for structure code if the attribute controls or is controlled by another attribute.

S/NAME — specifies column heading.

select list — a list of items created with one of the list commands.

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.

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 data base 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.

Notes

B Default Attribute Definition Items

When an account is created, several attribute defintion 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)

MD.....	D/CD	AMC	S/NAME.....	STRUCT....	CONVERSION.....	CORRELATIVE..	TP	MAX
*A0	A	00					L	10
*A1	A	01					L	10
*A10	A	10					L	10
*A11	A	11					L	10
*A12	A	12					L	10
*A13	A	13					L	10
*A14	A	14	Default				T	15
*A15	A	0	Help				T	15
*A16	A	16	Links				T	15
*A17	A	17					T	14
*A18	A	18						14
*A2	A	02						10
*A3	A	03						10
*A3DV	A	3						10
*A4	A	04						10
*A4DV	A	4						
*A5	A	05						10
*A6	A	06						10
*A6DV	A	06						
*A7	A	07						0
*A8	A	08						0
*A9	A	09						10
*A9998	A	9998						4
*A9999	A	9999						10
*A9DV	A	0						8
0	A	0	NAME					15
1	A	1	D/CD					3
10	A	10	MAX					3
2	A	2	AMC					2
3	A	03	S/NAME					10
4	A	04	STRUCT					10
5	A	05	CONVERSION					
8	A	8	CORRELATIVE					
9	A	09	TP					
A/AMC	A	02						
A1	A	01						2
A2	A	02						10
A3	A	03						25
A4	A	04						10
A5	A	05						0
D/CODE	A	1	CODE					
D/CODE1	A	1	CODE					1
D/SIZE	A	9999						4
F/BASE	A	02						2
F/LINKS	A	16						
F/MOD	A	03						16

Table B-1. Default Attribute Definition Items (2 of 2)

```
MD........ D/CD AMC S/NAME..... STRUCT.... CONVERSION..... CORRELATIVE   TP MAX

F/REALLOC  A     13                                                     L   7
F/SEP      A     04                                                     R   8
L/RET      A     05                                                     L   5
L/UPD      A     06                                                     L   5
S/AMC      A     04                                                     L   2
S/NAME     A     03                                                     L  20
V/CONV     A     07 CONVERSIONS                                         L  30
V/CORR     A     08 CORRELATIVE                                         L  25
V/EDIT     A     12                                                     L  10
V/MAX      A     10 MAX                                                 L   3
V/MIN      A     11                                                     L   3
V/TYP      A     09 TP                                                  L   2

58 items listed.

:
```

Notes

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.

ASSEMBLY File

Attribute 8 (V/CORR) specifies attribute 2 in the ASSEMBLY file as the field that contains the key to the product structure.

```
:CT  DICT  ASSEMBLY  *

  ASSEMBLY          1                  2
001 D             001 A              001 A
002 338861        002 1              002 2
003 3             003 DESCRIPTION    003 SUB-ASSEMBLY
004 1             004                004
005               005                005
006               006                006
007               007                007
008 V;;2          008                008
009 L             009 T              009 L
010 10            010 20             010 15

:CT  ASSEMBLY  *

    A100
001 Finished product
002 A10]A11]A12
    A10
001 Component A
002
    A11
001 Component B
002 A13]A14]
    A12
001 Component C
002
   A13
001 Raw material for B
002
    A14
001 Raw material for B
002
```

A100 is made up of three sub-items: A10, A11, and A12

A11 is made up of two sub-items: A13 and A14.

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
```

```
003 3114 Paradise Drive
004 San Fernando
005 CA
006 91340
007 8185551234
```

DICT Section

The following items make up the dictionary section of the CUSTOMERS file.

```
    1
001 A
002 1
003 Company
004
005
006
007 MCT
008
009 L
010 20

    2
001 A
002 2
003 Contact
004
005
006
007
008
009 L
010 20

    3
001 A
002 3
003 Address
004
005
006
007
008
009 L
010 20

    4
001 A
002 4
003 City
004
005
006
007 C4,' ';5 6
008
009 L
010 20

    5
001 X
002 5
003 State
004
005
006
007
008
009 L
010 2

    6
001 X
002 6
003 ZIP
004
005
006
007
008
009 L
010 5

    7
001 A
002 7
003 Phone Number
004
005
006
007 MR(###/=== ####)
008
009 R
010 12

    ADDRESS
001 A
002 3
003 Address
004
005
006
007 ML(#20)
008
009 L
010 20

    CITY
001 A
002 4
003 City
004
005
006
007 ML(#20)
008
009 L
010 20
```

```
    COMPANY
001 A
002 1
003 Company Name
004
005
006
007 ML(#30)
008
009 L
010 30
```

```
    NAME
001 A
002 2
003 Contact
004
005
006
007 ML(#20)
008
009 L
010 20
```

```
    PHONE
001 A
002 7
003
004
005
006
007 MR(###/###
####)
008
009 R
010 12
```

```
    STATE
001 A
002 5
003 State
004
005
006
007 ML(#5)
008
009 L
010 5
```

```
    ZIP
001 A
002 6
003 ZIP
004
005
006
007 MR(#5)
008
009 L
010 5
```

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]8123]7001]7015]7017]
    6032]6068]0005]2025]8036
004 2]1]1]1]2]1]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
006 O]O]O]O]O]O]O]O]O]O]O]O]O]O]O
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]6068]5011]5015
004 2]1]1]1]1]3
005 8892]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

    1683
001 Service Office Products
002 8988
003 7345]7331]8123]6032]6068]0005]2025]8036]4241]5003]
    5005
004 3]1]3]1]1]1]2]4]1]1]1
005 8892]8892]8892]8892]8892]8892]8892]8892]8892]8892]
    8892
006 C
```

```
007 Shelby
008 3114 Paradise Drive
009 91340
010 Net 30
011 SOP TRUCK
012 8185551234

    1686
001 Quality Lighting Products
002 9000
003 7056]7055]7066]7065]2025]6024]6032]0005]3007]3035]
    3004]5003]8123
004 2]2]1]1]4]1]2]1]1]1]1]1]1
005 8989]8989]8989]8989]8989]8989]8989]8989]8989]8989]
    8989]8989]8989
006 O]O]O]O]O]O]O]O]O]O]O]O
007 Mat
008 5454 W. Mariposa
009 91710
010 Net 10
011 Truck
012 3015551234

    1687
001 Service Office Products
002 8934
003 7056]7055]7066]7065]2025
004 1]1]1]1]3
005 8903]8903]8903]8903]8903
006 C
007 Lee
008 4512 Orange
009 92222
010 Net 30
011 SOP TRUCK
012 7145551234

    1696
001 Universal Copiers
002 8949
003 7345]7331
004 3]1
005 8920]8920
006 O
007 Marina
008 211 Westgate
009 91832
010 Net 30
011 UPS
012 2135551234
```

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]MR2$(#12)
008  A4*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;/]MR2$(#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
```

```
     1
001  A
002  1
003  Company Name
004
005
006
007  MCT
008
009  L
010  20

     COMPANY
001  A
002  1
003  Company Name
004
005
006
007  MCT
008
009  U
010  20

     COMPANY.L
001  A
002  1
003  Company Name
004
005
006
007  MCT]ML(#20)
008
009  L
010  20

     2
001  A
002  2
003  Invoice]Date
004
005
006
007  D2/
008
009  L
010  9
```

```
    DATE
001 A
002 2
003 Invoice]Date
004
005
006
007 D2/
008
009 L
010 9

    3
001 X
002 3
003 Product]Number
004
005
006
007
008
009 L
010 10

    4
001 A
002 3
003 Description
004
005
006
007 MCT
008 TPROD.NO;C;1;1
009 L
010 25

    PRICE.L
001 A
002 3
003 Price
004
005
006
007 MR2$(#10)
008 TPROD.NO;X;2;2
009 R
010 10

    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 5
003 Delivery]Date
004
005
006
007 D2/
```

```
008
009 L
010 9

    DEL.DATE
001 A
002 5
003 Delivery]Date
004 D;PROD.NO
005
006
007 D2/
008
009 L
010 9

    FLAGALL
001 A
002 6
003 Status
004
005
006
007 P(C);(O)
008
009 L
010 1

    NAME
001 A
002 7
003 Contact
004
005
006
007
008
009 L
010 20

    NAME.L
001 A
002 7
003 Contact
004
005
006
007 ML(#20)
008
009 L
010 20

    ADDRESS
001 A
002 8
003 Address
004
005
006
007
008
009 L
010 20

    ADDRESS.L
001 A
002 8
003 Address
004
005
006
007 ML(#20)
008
009 L
010 20

    CITY
001 A
002 9
003 City
004
005
006
007
008 A9["1","5"](TZIP;X;1;1)
009 L
010 20

    CITY.L
001 A
002 9
003 City
004
005
006
007 ML(#20)
008 A9["1","5"](TZIP;X;1;1)
009 L
010 20

    STATE
001 A
002 9
003 State
```

```
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
007
008 ML(#5)
009 L
010 5

    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 MR(###/### ####)
008
009 R
010 12

    NO-DET
001 A
002 34
003
004
005
006
007 MR2$(#10)
008 A4*3(TPROD.NO;X;2;2)
009 L
010 0
```

```
    TEN
001 A
002 34
003 \
004
005
006
007 C;""
008 A;(NI)/"10"
009 L
010 0

    MESSAGE
001 A
002 50
003
004
005
006
007
008
009 L
010 20

    CUMULATIVE
001 A
002 90
003 \
004
005
006
007 B;RUNNING.TOTAL]MR2$,(#10)
008 A4*3(TPROD.NO;X;2;2)
009 L
010 0

    TOT.PRICE
001 A
002 90
003 Tot Price
004
005
006
007 MR2$(#10)
008 A$(4*3(TPROD.NO;X;2;2))
009 R
010 10

    7
001 A
002 91
003 Ext Price
004
005
006
007 MR2$(#10)
008 A4*3(TPROD.NO;X;2;2)
009 R
010 10

    EXT.PRICE
001 A
002 91
003 Ext Price
004
005
006
007 MR2$(#10)
008 A4*3(TPROD.NO;X;2;2)
009 R
010 10

    NEXT.PRICE
001 A
002 91
003 Ext Price
004
005
006
007
008 A4*3(TPROD.NO;X;2;2)
009 L
010 10

    DESC
001 A
002 991
003 Description
004 D;PROD.NO
005
006
007 MCT
008 F;3(TPROD.NO;C;1;1)
009 U
010 15

    DESC.L
001 A
002 991
003 Description
004
005
006
007 MCT]ML(#25)
```

```
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
```

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

    6065
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
002 990

    7331
001 Tomato Juice
002 960

    7345
001 Apple Juice
002 1540
```

```
    8036
001 Cookies
002 670

    8123
001 Crackers
002 1030
```

DICT Section

The following items make up the dictionary section of the PROD.NO file.

```
    1
001 A
002 1
003 Description
004
005
006
007 MCT
008
009 L
010 20

    2
001 A
002 2
003 Price
004
005
006
007 MR2$
008
009 R
010 10
```

```
    DESC
001 A
002 1
003 Description
004
005
006
007 MCT
008
009 L
010 20

    PRICE
001 A
002 2
003 Price
004
005
006
007 MR2$
008
009 R
010 10
```

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
```

Notes

Index

- C -

- D -

- N -

- O -

- T -

- U -

- V -

- W -

- X/Y/Z -

Ultimate
THE ULTIMATE CORP.

ID
Number _____
for Ultimate use only

Problem Identification Form

Name	Phone Number ( )	System Number	Date

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:

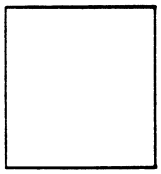
Name: _____ System Number: _____

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
THE ULTIMATE CORP.

Reader Comment Form

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.

Name	Phone Number ( )	System Number
Name of Manual	Document Number	Date
Comments:		

FROM:

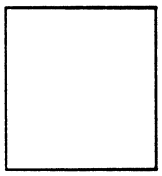
Name: _____ System Number: _____

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
THE ULTIMATE CORP.

Suggestion Form

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.

Name	Phone Number ( )	System Number	Date
Company Name	Hardware Platform: (manufacturer, model no.)		
Dealer Name	Host O/S and revision		
Suggestion:			

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.

717 Ridgedale Avenue
East Hanover, NJ 07936
201/887 9222
FAX: 201/887 9546