

NEC

Reality v10.0

English Utilities and Printing

NECSWS.COM

\Orchestrating a brighter world

Document control

Software Version	Document Status	Document Revision	Issue Date	Reason for Change
v10.0	Published	v0.1	June 2003	Final draft

Table of Contents

Section 1: About this guide	4
1.1 Purpose	4
1.2 Terminology change	4
1.3 References	4
1.4 File specifications	5
1.5 Selection criteria	5
1.6 Standard options	6
1.7 Conventions	6
Section 2: English and related commands	8
2.1 English commands	8
2.2 English-related commands	9
Section 3: Conversion codes	11
3.1 A Code - Algebraic Functions	11
3.2 Operands	11
3.3 Arithmetic operators	13
3.4 Relational operators	13
3.5 Logical operators	13
3.6 Other operators	14
3.7 Format codes	14
Section 4: Appendix A Output Format Modifiers	24
Section 5: Appendix B Output Specifications	27
Section 6: Appendix C Data Definition Items	29

Section 1: About this guide

1.1 Purpose

English is a powerful inquiry language used for data retrieval and report generation from a Reality or Reality database.

This guide lists the verbs that belong to or are closely associated with the English inquiry language. Other elements of this language are summarised.

A summary of the processing codes used in English dictionaries, which provide definitions of the data elements used within English inquiries, is then given.

It will be useful to experienced Reality users, Pick users migrating to Reality and to new users exploring the system.

Caution

It is outside the scope of this guide to fully define each element in context. If in doubt, check the detailed description in the English Reference Manual.

1.2 Terminology change

In English reference documentation for dictionaries, the following changes have been made:

Data Definition Item replaces *Attribute Definition Item*.

'Pre-processor' (or 'Attribute 8 codes') replaces 'correlatives'.

'Input / output conversions' (or 'Attribute 7 codes') replaces 'conversions'.

1.3 References

Beginner's Guide to English

This tutorial is a good introduction to the commonly used features of English, and to the creation of dictionaries.

English Reference Manual

This gives more detailed descriptions of all elements of English, with examples.

Editor Reference Manual

Screen Editor Reference Manual

Describe the editors that are used to create data definition items in English dictionary files.

General Utilities and Printing

This includes details of special utilities including some that can be executed from English dictionaries for.

Proc Reference Manual

This describes the Proc programming language. Proc enables programs to be written to execute one or more English commands, of any complexity, via a one-word command, and also allows interaction with the user. Procs can be run immediately after creation as no compilation is necessary.

DataBasic Reference Manual

This describes DataBasic, a more powerful programming language than Proc that requires compilation. English commands can be executed via DataBasic 'PERFORM's.

1.4 File specifications

The term filespec indicates a file that can be specified as follows:

{*DICT*} *{/account/}file{,data-sect}*

1.5 Selection criteria

Selection criteria in English commands consist of item selection clause(s) and/or attribute selection clause(s). Limits items processed to those satisfying condition(s) in either case.

Item selection clauses consist of:

relational-operator value-list

where relational-operator must be one of:

= (or synonym **EQ**)

(or synonym **NE**)

> (or synonym **GT**)

< (or synonym **LT**)

<= (or synonym **GE**)

>= (or synonym **LE**)

BETWEEN (must precede two values separated by **AND**).

Value-list is as defined below for attribute selection, except that each value or value string is enclosed in single quotes and *relational-operator* is mandatory.

These clauses generally follow *filespec*, as they select item-ids satisfying condition(s) specified.

Attribute selection clauses consist of:

[IF | WITH]{[NO | NOT] [EVERY | EACH]} *attribute-name { {relational-operator} value-list }*

These select for processing all items that satisfy conditions specified. **IF** is synonym for **WITH** and is mandatory. **NO** is synonym for **NOT** and **EVERY** is synonym for **EACH**. At most, one of these can be used. **EVERY** specifies that all values must satisfy conditions specified. **NO** specifies that all of the attribute's values must be null (if **NO** or **NOT** specified, relational-operator and value-list are redundant).

In attribute selection clauses, value-list consists of:

value-string { *{logical-operator}* *value string* }

where value-string consists of a character string, optionally including special characters ^ (to match any character), [(at left of string, matches any number of characters preceding value) and/or] (matches any number of characters to right of value, or matches any string when used within the string). Each value string must be enclosed in double quotes. Logical-operator is **AND** or **OR**. If omitted, **OR** is implied. **OR** means select every item satisfying any condition. **AND** means select items satisfying every condition.

Multiple selection criteria can be combined with **AND** or **OR** (**OR** implied if neither specified). **AND** connectives are processed before **ORs**.

1.6 Standard options

The following options are standard with most English commands, and perform the function described below, unless otherwise specified with the command.

Syntax elements	Description
B	Suppresses initial linefeed of output.
C	Suppresses column headings, default page headings and final summary line. (Equivalent to COL-HDR-SUPP modifier.)
D	Suppresses all detail and blank lines - that is, outputs lines containing subtotals and accumulated total only. (Equivalent to DET-SUPP .)
H	Suppresses default page heading and final summary line. (Equivalent to HDR-SUPP/SUPP .)
I	Suppresses item-id column. (As ID-SUPP .)
N	Suppresses automatic paging. (As NOPAGE .)
P	Outputs report to current spooler assignment (usually printer). (As LPTR modifier.)
R	Suppresses 'Item not on file' messages.

1.7 Conventions

This guide uses the following conventions:

Example	Meaning
TEXT	Bold text represents characters typed exactly as shown.
{param}	Braces enclose options and optional parameters.

Example	Meaning
...	Indicates that preceding parameters can be repeated as many times as necessary.
<i>text</i>	Italic text indicates parameters you must supply or references to other documents.
[param param]	Square brackets containing parameters separated by vertical lines indicate that you must select at least one of these parameters.
CTRL+X	Two (or more) key names joined by a plus sign (+) indicate a combination of keys, where the first key(s) must be held down while the second (or last) is pressed.
RETURN	Small capitals show key names.
X'nn'	This denotes a hexadecimal value.

Section 2: English and related commands

2.1 English commands

The verbs listed below form the English command language, and use the common English syntax:

verb {file-modifier} filespec {item-list} {selection-criteria} {sort-criteria} {output-specification} {format-specification} {(options)}

For more details of each command, see *TCL Quick Reference Guide* or *English Reference Manual*.

Command	Description
BSELECT	Generates a select list derived from attributes in output specification for selected items.
CHECK-SUM	Calculates a checksum for selected file contents.
COUNT	Counts selected items.
DEFINE-INDEX	Defines selection and sort criteria for an index.
EDELETE	Deletes selected file items.
ESEARCH	Generates select list of items containing or not containing specified character strings.
HASH-TEST	Generates file hashing statistics based on test modulo.
I-DUMP	Dumps file contents.
ISTAT	Generates file hashing statistics based on existing modulo.
LIST	Output formatted listing (report) of selected file contents.
LIST-ITEM	Outputs all attributes of selected items.
LIST-LABEL	Output selected file contents in label format.
LIST-SPREAD	Outputs formatted listing of file for spreadsheet use.
REFORMAT	Outputs formatted data to a file.
S-DUMP	Dumps file contents in sorted order.
SELECT	Generates active list of selected items.
SORT	Outputs formatted listing (report) of file contents in sorted order.
SORT-ITEM	Outputs all attributes of selected file items in sorted order.
SORT-SPREAD	Outputs formatted listing of file, in sorted order, for spreadsheet use.
SREFORMAT	Outputs formatted data to a file, in sorted order.
SSEARCH	Generates sorted select list of items containing or not containing specified character strings.
SSELECT	Generates a sorted active list of selected file items.
ST-DUMP	Dumps selected file items to tape, in sorted order.
STAT	Generates total sum, average and count for numeric data in specified attribute.

Command	Description
SUM	Generates total sum for numeric data in specified attribute.
T-DUMP	Dumps selected file items to tape.
T-LOAD	Loads selected items from tape.

2.2 English-related commands

The following do not use standard English syntax but are associated with English. For example, many process item lists that can be used with English. For more information, see TCL command descriptions.

Command	Description
AND-ITEMS	Generates select list from contents of items via logical AND.
AND-LISTS	Generates select list from two or more saved lists via logical AND.
CLEAR-	Clears special options set by OPTION SET-OPTION, including some that modify English functionality.
COPY-LIST	Copies a saved list to another saved list or to a file item.
DATE-FORMAT	Changes date format (numerical in some English reports) for duration of logon session on local port. See also SET-OPTION.
DELETE-LIST	Deletes saved POINTER-FILE item list.
EDIT-LIST	Allows processing of stored POINTER-FILE item lists via EDITOR commands.
FORM-LIST	Generates select list from attributes of a specified item.
GET-LIST	Retrieves a saved item list (from POINTER-FILE).
LIST4	Lists file items in 4-column format.
LISTCONN	Lists English connectives from MD or other file
LISTDICTS	Lists data definitions in MD or in dictionary of specified file.
LISTF2, 3, 4	Lists files defined from current account in 2, 3, or 4 columns.
LISTFILES	Lists files, data sections and indexes defined from current account or specified file's dictionary.
LISTQFILES	Lists synonym files in current or named account.
LOAD-ALPHA	Loads alternative character sets.
NEW-GET-LIST	Retrieves a saved item list from a file.
NEW-SAVE-LIST	Saves an active list to a file item.
NEW-SORT-LIST	Sorts (and optionally further processes) a list saved in a file item.
NSELECT	Generates select list of items in active select list but not in file.
OR-ITEMS	Generates select list from contents of items via logical OR.

Command	Description
OR-LISTS	Generates select list from two or more save lists via logical OR.
QSELECT	Generates select list from attributes of items.
SAVE-LIST	Saves active list (in POINTER-FILE).
SEARCH	Generates a select list of items containing or not containing specified character strings.
SELECT-INDEX	Generates a select list of selected file index entries.
SET-ALPHA	Sets a specified range of hexadecimal values to be alphabetic.
SET-COLL	Loads specified collation table into the table file.
SET-DEC	Defines character used as decimal separator in numeric conversions by requesting process until logoff.
SET-MONEY	Defines character used in numeric conversions as monetary denomination symbol for your port until logoff.
SET-OPTION	Sets special options during current session of executing process.
SET-THOUS	Defines character used by process in numeric conversions as thousands separation symbol.
SORT-LIST	Sorts (and optionally further processes) a saved list in POINTER-FILE.
SORT2, 3, 4	Lists item-ids in a file in sorted order in 2, 3 or 4 columns.
XOR-ITEMS	Generates a select list comprising unique attributes within merged items.
XOR-LISTS	Generates a select list comprising unique entries within merged lists.
XSELECT	Generates a select list of item-ids that are in a file but not in an active list.

Section 3: Conversion codes

All of the following codes can be used either as pre-processor (attribute 8) or input/output (attribute 7) codes, unless stated otherwise.

3.1 A Code - Algebraic Functions

Processes data in an algebraic format.

AE;expr

or

A{n}{;expr}

AE evaluates expression using integer and fraction parts of stored numbers. **A** performs expression on values with embedded decimal point and converts value to scaled integer; only uses integer parts of stored numbers (unless scaling factor n is included).

Syntax elements	Description
<i>n</i>	Scaling factor in range 1-6.
<i>expr</i>	Operands, operators, conditional statements and special functions:

3.2 Operands

attr-num{**R**{**R**}}

Returns value of attribute. Reserved attribute numbers mean:

0 item-id

9998 ordinal number of item

9999 item size in bytes

R	If one attribute in expression has only one value, it will be repeated to match the number of values of other attribute (default is to use first value only).
RR	If one attribute in expression has only one subvalue, it will be repeated to match the number of subvalues of other attribute (default is to use first subvalue of first value of attribute).

{-}D

Returns system date (internal format).

I(expr)

Returns integer part of expression.

"string"

Returns literal string or numeric constant.

{-}LPV

Returns previous value transformed by format code.

N(attr-name){R{R}}

Returns value as defined by Data Definition Item.

R	Returns each value individually (default is first value of attribute).
RR	Returns each subvalue individually (default is first subvalue of first value of attribute).

{-}NA

Returns number of attributes in item.

{-}NB

Returns current break level count (1 = lowest break level; 255 = GRAND-TOTAL line).

{-}ND

Returns item count (detail lines) since last control break.

NF

Returns complete file specifier.

NFA

Returns account name part (if any) of entered file specifier.

NFD

Returns "DICT " if dictionary access.

NFF

Returns entered file specifier with DICT and sectionname strings, if any, removed.

NFN

Returns file name part of entered file specifier.

NFS

Returns section name part of entered file specifier.

{-}NI

Returns item count.

{-}NL

Returns item length.

{-}NS

Returns subvalue count.

{-}NU

Returns date of last update.

{-}NV

Returns value count.

R(expr1,expr2)

Returns remainder of first expression divided by second.

S(expr)

Returns sum of all the values generated by expr.

string[s,n]

Returns substring of n characters, starting at character s. (s and n can be expressions.)

{-}T

Returns system time (internal format).

{-}V

Returns previous value transformed by format code.

{-}-

Changes default sign (+) of value.

3.3 Arithmetic operators

+	Sum of operands
-	Difference between operands
*	Product of operands
/	Quotient (integer value) of operands

3.4 Relational operators

= or EQ	Equal to
< or LT	Less than
> or GT	Greater than
<= or LE	Less than or equal to
>= or GE	Greater than or equal to
# or NE	Not equal to

3.5 Logical operators

AND	Evaluates to 1 if both expressions are true, otherwise 0.
OR	Evaluates to 1 if at least one of the expressions is true, otherwise 0.

3.6 Other operators

:

Concatenation operator.

IF expr **THEN** expr **ELSE** expr

Gives A code conditional capabilities. Expressions must evaluate to true or false and statement is string or numeric value.

3.7 Format codes

AE;operand(format-code[format-code]...)

or

A{*n*}; operand(format-code[format-code]...)

Converts stored values to external (output) format.

Format code(s) can be any of those listed subsequently in this chapter. Separator between format codes is value mark (CTRL+]).

C Code - Concatenation

C{;} *n*{*xn*}...

Concatenates attributes, literals in quotes and previous results with optional single-character separators. Concatenation not inverted on input - conversion applied as if output.

Syntax elements	Description
;	Optional - allows SMA compatibility.
<i>n</i>	Attribute number (AMC) or literal in backslashes or single or double quotes or asterisk (*) for last value generated by previous operation.
<i>x</i>	Single character (or SPACE) to use as separator. Semicolon means no separator.

D Code - Date

D{*p*}{*n*}{*xm*}{*s*}

Converts dates from internal to external format. D on its own gives the format dd mmm yyyy (two digit day, three letter month, four digit year).

Syntax elements	Description
<i>p</i>	special processing operator:
	AM Returns administrative month number.
	AW Returns administrative week number.
	AY Returns administrative year.
	D Returns number of day of month.
	I Converts from external to internal format.
	J Returns day of year in range 1 to 366.
	M Returns month number in range 1 to 12.

Syntax elements	Description												
	<table> <tr> <td>MA</td><td>Returns month name in initial capital letters.</td></tr> <tr> <td>Q</td><td>Returns quarter number in range 1 to 4.</td></tr> <tr> <td>S</td><td>Returns data as year, followed by month, followed by day.</td></tr> <tr> <td>W</td><td>Returns day number (Monday is 1).</td></tr> <tr> <td>WA</td><td>Returns day name in initial capital letters.</td></tr> <tr> <td>Y</td><td>Returns year.</td></tr> </table>	MA	Returns month name in initial capital letters.	Q	Returns quarter number in range 1 to 4.	S	Returns data as year, followed by month, followed by day.	W	Returns day number (Monday is 1).	WA	Returns day name in initial capital letters.	Y	Returns year.
MA	Returns month name in initial capital letters.												
Q	Returns quarter number in range 1 to 4.												
S	Returns data as year, followed by month, followed by day.												
W	Returns day number (Monday is 1).												
WA	Returns day name in initial capital letters.												
Y	Returns year.												
<i>n</i>	number of rightmost digits to be printed in year field (range 0 to 4 , default 4).												
<i>x</i>	nonnumeric separator (excluding minus/hyphen and system delimiters) used in data.												
<i>m</i>	number of data segments to skip before reaching date, where x separates segments.												
<i>s</i>	nonnumeric character to separate day, month and year (default is SPACE). Gives format with numerics for day, month and year - order depends on DATE-FORMAT (TCL command) setting or database configuration.												

D1 and D2 Codes - Controlling and Dependent Attributes

Associate controlling and dependent attributes allowing logical grouping of multivalued attributes. Can be followed by other format codes, separated by VMs (CTRL+]). Only work as pre-processor (attribute 8) codes.

D1;amcd{;amcd }...

Defines controlling associative attribute (by pointing to dependent associative attributes).

Syntax elements	Description
<i>amcd</i>	AMC of dependent associative attribute.

D2;amcc

Defines dependent associative attribute (by pointing to controlling associative attribute).

Syntax elements	Description
<i>amcc</i>	AMC of controlling associative attribute.

F Code - Mathematical Functions

Performs specified operations on one or more operands on a stack (holding up to 80 entries). All operators pop entries used (except pop operators) and push result into entry 1.

F{*n*};elem{;elem}...

Only uses integer parts of stored numbers (unless scaling factor is included). Uses Reverse Polish Notation and non-standard order of operation (processes stack entry 1 before stack entry 2).

Syntax elements	Description
<i>n</i>	Scaling factor in range 1 to 6 .
<i>elem</i>	Valid operators:

attr-num{**R**{**R**}}{(format-code)}

Syntax elements	Description
R	Indicates first value of attribute to be used repeatedly when evaluating multivalued attribute elsewhere in calculation.
RR	Indicates first subvalue to be used repeatedly when evaluating subvalued attribute elsewhere in calculation.

'literal ' or "literal " Literal

C <i>n</i>	Returns constant where <i>n</i> is string constant.
D	Returns current date.
LPV	Returns previous value.
NA	Returns number of attributes.
NB	Returns current break level (1 = lowest break level; 255 = GRAND-TOTAL line).
ND	Returns current break level (1 = lowest break level; 255 = GRAND-TOTAL line).
NF	Returns complete file specifier.
NFA	Returns account name part of entered file specifier.
NFD	Returns "DICT " if dictionary access.
NFF	Returns entered file specifier with DICT and section-name strings, if any, removed.
NFN	Returns file name part of entered file specifier.
NFS	Returns section name part of entered file specifier.
NI	Returns item count.
NL	Returns item length.
NS	Returns subvalue count.
NU	Returns date of last update.
NV	Returns value count.

P	Returns duplicate of entry 1.
T	Returns time in internal format.
V	Returns previous value.
"	Returns duplicate of entry 1.

The above push a value onto the stack. Other operators are the arithmetic ($\{n\}$, $/$, **R**, **+**, **-**, **I**, **S**), miscellaneous ($_$, \wedge , (format-code), $:$, $[]$), relational ($=$, $<$, $>$, $[]$, $\#$) and logical (**&**, **!**). Some of these are described under **A Code**.

FS;elem {;elem}...

Only uses integer parts of stored numbers. Uses Reverse Polish Notation and SMA Standard order of operation (processes stack entry 2 before 1). *Elem* is any valid operator (see above).

FE;elem {;elem}

Uses integer and fraction parts of stored numbers. Uses Reverse Polish Notation and SMA Standard order of operation (processes stack entry 2 before 1). *Elem* is any valid operator (see above).

G Code - Group Extraction

G{*m*} *xn*

Extracts one or more contiguous segments (separated by specified character) from attribute value.

Syntax elements	Description
<i>m</i>	Number of separation characters to count, from the start of the attribute value (if <i>m</i> positive), or from the end of the attribute value (if <i>m</i> negative), to position at the start of the segments to be extracted (default is zero).
<i>x</i>	Separation character (any nonnumeric except system delimiter).
<i>n</i>	Number of separation characters to count, from the start of the segments to be extracted (if <i>n</i> positive), or the end of the attribute value (if <i>n</i> negative), to position at end of segments to be extracted.

L Code - Length

L{*{m,} n*}

Returns length of value in characters (if *m* and *n* omitted). Otherwise, returns complete value if of defined length.

Syntax elements	Description
<i>n</i>	Returns value if length $\leq n$.
<i>m</i>	Returns value if length $\geq m$.

MC Code - Mask Character

MCs

Performs character conversions, retaining delimiters except for MCP and MCPN. Pre-processor (attribute 8) code only, except as indicated. See **LOAD-ALPHA** (TCL cmd) for information on additional alphabets.

s Suffix operator code:

0	Extracts standard alphabetic character only.
1	Extracts characters defined by primary added alphabet only.
2	Extracts characters defined by secondary added alphabet only.
A	Extracts only alphabetic characters (from any alphabet defined).
/A	Extracts only nonalphabetic characters.
AB{S}	Converts ASCII to binary; S suppresses spaces.
AX	Converts ASCII to hex.
B	Extracts only alphabetic and numeric characters.
/B	Extracts characters that are not alphabetic or numeric.
BA	Converts binary to ASCII.
BX	Converts binary to hex.
C;x;y	Changes all occurrences of string x to string y.
DR	Converts decimal to Roman numerals. (Can be used as input/output conversion.)
DX	Converts decimal to hex. (Can be used as input/output conversion.)
L	Converts upper case letters to lower case.
N	Extracts only numeric characters.
/N	Extracts only nonnumeric characters.
NP{c}	Converts paired hex digits preceded by tilde (~) or c to ASCII.
P{c}	Converts all nonprintable characters (X'00'- X'1F', X'80'- X'FA') to tildes (~) or c.
PN{c}	Same as MCP, but inserts two-character hex after tilde (~) or c.

RD	Converts Roman numerals to decimal. (Can be used as input/output conversion.)
T	Converts upper case text to initial capitals.
U	Converts lower case letters to upper case.
XA	Converts hex to ASCII.
XB{S}	Converts hex to binary; S suppresses spaces.
XD	Converts hex to decimal. (Can be used as input/output conversion.)

MX is synonym for **MCAX**

MCY is synonym for **MCXA**

MCD is synonym for **MCDX**

MCR is synonym for **MCRD**

MCX is synonym for **MCXD**

MD Code - Mask Decimal without Justification

MD{*n*{*m*}}{**Z**}{,}{**£**}{**\$**}{*ix* }{*c* }

Transforms integers for display by scaling and inserting symbols. Input conversion works on numbers with decimal and/or thousands separator only.

Syntax elements	Description						
<i>n</i>	Number of places after decimal separator in range 0 to 9 (default is 0 - no decimal separator).						
<i>m</i>	Descaling factor. Default is <i>n</i> . If <i>m</i> is greater than <i>n</i> , values are rounded to <i>n</i> digits. If <i>ix</i> is used without Z and £ or \$, <i>m</i> is mandatory.						
Z	Suppresses leading zeros (except one preceding decimal point), unless data is zero, when there is no output.						
,	Inserts thousands separators.						
£ or \$	Prefixes value with currency symbol.						
<i>ix</i>	Aligns £ or \$ by creating <i>i</i> blank columns; <i>x</i> is any nonnumeric filler character (including SPACE).						
<i>c</i>	Applies credit character to negative values: <table> <tr> <td>-</td><td>minus sign.</td></tr> <tr> <td>C</td><td>literal CR.</td></tr> <tr> <td><</td><td>Encloses with angle brackets.</td></tr> </table>	-	minus sign.	C	literal CR.	<	Encloses with angle brackets.
-	minus sign.						
C	literal CR.						
<	Encloses with angle brackets.						

MK - Mask Metric

MKn

Allows the display of large numbers using letters for power of 10. K, M and G stand for 10^3 , 10^6 and 10^9 . Input conversion is not inverse of output conversion.

Syntax elements	Description
<i>n</i>	Field size - number of characters to be output including letter and any minus sign.

ML and MR Codes - Mask Decimal with Justification

ML{*n*{*m*}}{**Z**}{,}{*c* }{**£**}{**\$**}{*fm* }

or

MR{*n*{*m*}}{**Z**}{,}{*c* }{**£**}{**\$**}{*fm* }

Formats numbers with left and right justification respectively. Input conversion works on numbers with decimal and/or thousands separators only.

ML justifies numbers to left; **MR** to right.

Syntax elements	Description										
<i>n</i>	Number of places after decimal separator in range 0 to 9 (default is 0 - no decimal separator).										
<i>m</i>	Decaling factor. Default is <i>n</i> .										
Z	Suppresses leading zeros (except one preceding decimal separator).										
,	Thousands separators.										
£ or \$	Prefixes value with currency symbol.										
<i>c</i>	Uses credit character: <table border="1"> <tr> <td>C</td><td>Appends literal CR to negative values.</td></tr> <tr> <td>D</td><td>Appends literal DB after positive values.</td></tr> <tr> <td>E</td><td>Encloses negative values with angle brackets.</td></tr> <tr> <td>M</td><td>Appends minus sign to negative values.</td></tr> <tr> <td>N</td><td>Suppresses embedded minus sign.</td></tr> </table>	C	Appends literal CR to negative values.	D	Appends literal DB after positive values.	E	Encloses negative values with angle brackets.	M	Appends minus sign to negative values.	N	Suppresses embedded minus sign.
C	Appends literal CR to negative values.										
D	Appends literal DB after positive values.										
E	Encloses negative values with angle brackets.										
M	Appends minus sign to negative values.										
N	Suppresses embedded minus sign.										
<i>fm</i>	Format mask comprising literal characters and format codes: <table border="1"> <tr> <td>#{<i>n</i>}</td><td><i>n</i> blanks used and overwritten by value as required.</td></tr> <tr> <td>*{<i>n</i>}</td><td><i>n</i> asterisks used and overwritten by value as required.</td></tr> <tr> <td>%{<i>n</i>}</td><td><i>n</i> zeros used and overwritten by value as required.</td></tr> <tr> <td>&{<i>n</i>}</td><td><i>x</i> is any of above format codes, currency symbol, space or literal; character after & fills columns without data.</td></tr> </table>	# { <i>n</i> }	<i>n</i> blanks used and overwritten by value as required.	* { <i>n</i> }	<i>n</i> asterisks used and overwritten by value as required.	% { <i>n</i> }	<i>n</i> zeros used and overwritten by value as required.	& { <i>n</i> }	<i>x</i> is any of above format codes, currency symbol, space or literal; character after & fills columns without data.		
# { <i>n</i> }	<i>n</i> blanks used and overwritten by value as required.										
* { <i>n</i> }	<i>n</i> asterisks used and overwritten by value as required.										
% { <i>n</i> }	<i>n</i> zeros used and overwritten by value as required.										
& { <i>n</i> }	<i>x</i> is any of above format codes, currency symbol, space or literal; character after & fills columns without data.										

MP Code - Masked Packed Decimal

MP

Converts packed decimals to unpacked decimals.

MT Code - Masked Time

MT{H}{S}

Converts time from internal format (number of seconds from midnight) to 12 or 24-hour format.

Syntax elements	Description
H	12-hour format (default is 24 hour).
S	Shows seconds (default is hours and minutes only).

P Code - Pattern Matching

P(*elem*){;(elem)}...

Returns - unaltered - any value matching one or more specified patterns (returns null if no match). Patterns can be combinations of numeric, alphabetic and literal strings. Inverse conversions just apply matching to values in command.

Syntax elements	Description
<i>elem</i>	Consists of one or more of the following pattern specifiers:
	<i>nN</i> <i>n</i> numeric characters.
	<i>nA</i> <i>n</i> alphabetic characters.
	<i>nX</i> <i>n</i> characters.
	' <i>literal</i> ' Literal.

R Code - Range Check

R*n,m*{;*n,m*}...

Returns - unaltered - any value falling within one or more specified ranges (returns null if not within any range). Inverse conversion just applies range check to values in command.

Syntax elements	Description
<i>n</i>	First integer of range (positive or negative).
<i>m</i>	Last integer of range (positive or negative); must be greater than or equal to <i>n</i> .

S Code - Substitution

S; *val-substitute*; *zero-substitute*

Specifies substitution elements for values: can be AMC of an attribute, literal in single quotes or asterisk (*) for value generated by previous operation. Input conversion applies substitution to values in command.

Syntax elements	Description
<i>val-substitute</i>	element to replace nonzero/null value.
<i>zero-substitute</i>	element to replace nonzero/null value.

T Code - Text Extraction

T{*m*,} *n*

Extracts character substring from value. Input conversion just applies text extraction to values in command.

Syntax elements	Description
<i>m</i>	Starting column number.
<i>n</i>	Number of characters to extract starting from column <i>m</i> . If <i>m</i> omitted, <i>n</i> specifies number of characters to extract starting from left of value (or from right if justification code is R).

Tfile Code - File Translation

T{[{**DICT** |*]}*filespec*};*c*{[*n*:*w*]};{*i-amc*};{*o-amc*};{*b-amc*}

Extracts data from another file.

Data value being processed is used as item-id in translation file. If *filespec* is omitted, value being processed must include translation file specification.

Input conversion must be explicitly specified if required.

Syntax elements	Description										
*	Specifies dictionary of translation file (alternative to DICT).										
<i>filespec</i>	Translation file.										
<i>c</i>	<p>Uses credit character:</p> <table> <tr> <td>C</td><td>If translate item does not exist or specified attribute in translate item is null, value is used unaltered.</td></tr> <tr> <td>I</td><td>Acts as C for output and V for input translation.</td></tr> <tr> <td>O</td><td>Acts as C for input and V for output translation.</td></tr> <tr> <td>V</td><td>Translate item must exist and specified attribute must not be null (or error message results).</td></tr> <tr> <td>X</td><td>Uses null if translate item does not exist or if specified attribute is null.</td></tr> </table>	C	If translate item does not exist or specified attribute in translate item is null, value is used unaltered.	I	Acts as C for output and V for input translation.	O	Acts as C for input and V for output translation.	V	Translate item must exist and specified attribute must not be null (or error message results).	X	Uses null if translate item does not exist or if specified attribute is null.
C	If translate item does not exist or specified attribute in translate item is null, value is used unaltered.										
I	Acts as C for output and V for input translation.										
O	Acts as C for input and V for output translation.										
V	Translate item must exist and specified attribute must not be null (or error message results).										
X	Uses null if translate item does not exist or if specified attribute is null.										

Syntax elements	Description
<i>n</i>	Value mark count for value in transtart file attribute or *. A * instead of n uses the same value in the target as in the primary file. If omitted, translation to multivalued field returns all values and multivalues, separated by spaces or w.
<i>w</i>	Character to delimit multivalues and subvalues if n is null (default is spaces).
<i>i-amc</i>	Attribute number for input translation. If omitted, no translation is applied. Specify 9999 to return size of target item.
<i>o-amc</i>	Attribute number for output translation. If omitted, no translation is applied. Specify 9999 to return size of target item.
<i>b-amc</i>	Attribute to be used (instead of o-amc) in a BREAK-ON/TOTAL output line.

U Code - User Exits

U xxxx

Executes special routine to process values. Do not call a user exit unless familiar with its action. See the manual *General Utilities and Printing* for details of utilities available.

Syntax elements	Description
xxxx	Hexadecimal identity of routine.

Section 4: Appendix A Output Format Modifiers

These optional English clauses modify the format of the report produced. They do not alter the effect of default or explicit output specifications.

COL-HDR-SUPP

Suppresses column headings, default page headings (page number, time and date) and summary line ("*n* ITEMS LISTED"). Equivalent to **C** option.

DBL-SPC

Double spaces lines of report (not label).

DET-SUPP

Suppresses detail output (all lines except subtotal and total lines) when used with **BREAK-ON** and **TOTAL** connectives. Equivalent to **D** option.

FOOTING "{text }{'opt {opt }...'}..."

Specifies footer text (printable characters); two successive single quotes are output as single quote. To separate options with a space (or other character), each must be enclosed in its own pair of quotes.

Options:

Option	Description
B	Replaces B in footer with last value of BREAK-ON attribute on page. Only works if BREAK-ON with B option also specified.
C{n}	Centres text (optionally to column number <i>n</i>).
D	Inserts system date in form: <i>dd mmm yyyy</i> .
F	Inserts file name.
I	Inserts current item-id. First item-id on page is inserted in header and last in footer.
L	Starts new line at point where L appears.
N	Suppresses automatic paging of terminal output.
P	Inserts current page number, left justified.
PP	Inserts current page number, right justified.
T	Inserts system time and date in form: <i>hh:mm:ss dd mmm yyyy</i> .

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

or

CAPTION "{text }{'opt {opt }...'}..."

Specifies text (printable characters) to replace default asterisks in cumulative total line at end of report.

Options:

Option	Description
L	Suppresses blank line preceding grand total line. Overrides U option.
P	Outputs grand total on separate page.
U	Puts underlines on line above accumulated totals. Overridden by L option.

HEADER "{text }{'opt {opt }...' }..."

or

HEADING "{text }{'opt {opt }...' }..."

Specifies heading text; two successive single quotes are output as one single quote. If options are to be output separated by spaces (or other character), enclose each option in its own pair of quotes.

Options:

Option	Description
B	Replaces B in footer with last value of BREAK-ON attribute on page. Only works if BREAK-ON with B option also specified.
C{n}	Centres text (optionally to column number <i>n</i>).
D	Inserts system date in form: <i>dd mmm yyyy</i> .
F	Inserts file name.
I	Inserts current item-id. First item-id on page is inserted in header and last in footer.
L	Starts new line at point where L appears.
N	Suppresses automatic paging of terminal output.
P	Inserts current page number, left justified.
PP	Inserts current page number, right justified.
T	Inserts system time and date in form: <i>hh:mm:ss dd mmm yyyy</i> .

HDR-SUPP or **SUPP**

Suppresses default heading and summary lines. Equivalent to **H** option.

ID-SUPP

Suppresses item-ids for **LIST** and **SORT** processors. Equivalent to **I** option.

LPTR

Sends report to current spooler assignment (usually printer), instead of to terminal. Equivalent to **P** option.

NOPAGE

Suppresses automatic pagination of terminal display. Equivalent to **N** option.

Section 5: Appendix B Output Specifications

This optional clause of an English command specifies attributes to be included in report/output and their sequence (but note that dependent attributes are always printed to the right of their controlling attribute).

Optionally they can restrict output to items with specified attribute matching specified print limiter. Also, **TOTAL**, **MIN**, **MAX** and **BREAK-ON** modifiers can be used as shown below.

Use of an output specification overrides any default attribute list.

TOTAL | MIN | MAX *attrib* {*print-limiter*}...

TOTAL keeps running total of numeric attribute and displays it at each break and for cumulative total.

MIN causes the minimum value of the attribute to be output at a control break.

MAX causes the maximum value of the attribute to be output at a control break.

If used with **BREAK-ON**, **TOTAL** displays intermediate totals at breaks. Can also use **GRAND-TOTAL** to specify text for display on last total line (see Appendix A).

Syntax elements	Description
<i>attrib</i>	name of attribute to be totalled.
<i>print-limiter</i>	Only values matching this are output. Has the form: { NOT } { <i>relational-operator</i> } " <i>value-string</i> " { <i>logical-connective</i> } { NOT } { <i>relational-operator</i> } " <i>value-string</i> "...

BREAK-ON *attrib* {"{*break-text*}{'opt {opt }...'}'..."} }

Generates a break line in report each time specified attribute changes value.

Syntax elements	Description
<i>attrib</i>	name of attribute to be monitored.
<i>break-text</i>	printable characters (except RETURN, LINEFEED, quotes and system delimiters).

Options:

Option	Description
B	Puts break value in HEADING or FOOTING (which must have been specified).
D	Suppresses break line if only one detail line since last break.
L	Suppresses blank line preceding break line. Overrides U option.
P	Starts new page after each break.

Section 5: Appendix B Output Specifications

Option	Description
R	Inhibits page break until all data associated with current break is output.
U	Puts underlines on line above accumulated totals if TOTAL specified. Ignored if L option also used.
V	Inserts value of control break attribute at this point in BREAK-ON label.

Section 6: Appendix C Data Definition Items

Data Definition Items (DDI) in a dictionary file, previously known as Attribute Definition Items, define the characteristics of a data file in terms of:

- default output: attributes with integer DDI item-ids from 1 upwards are output, unless output specified explicitly or default suppressed via ONLY, and provided integer sequence is unbroken.
- output format: how the associated attribute should be processed before sorting or selection (preprocessor (attribute 8) codes), and then before output (input/output (attribute 7) codes).
- associating attribute names with attribute numbers:

the DDI item-id is the name that identifies the associated attribute in an English command.

Any number of DDIs can be set up in a dictionary. The general structure of a DDI is given below, where 'attribute number' corresponds to line number of file:

Att.	Contents										
1	A , X or S . Any of these indicate that item is a DDI. X additionally means skip listing this attribute when item-id is an integer and would otherwise be output by default. If attribute 3 is null, S like A with attribute 3 containing just a backslash.										
2	A/AMC (Attribute Mark Count). Attribute number in item within data file. (Item-id of item is attribute 0). Numbers with special meanings are: 9998 ordinal number of item 9999 size of item in bytes.										
3	Column heading used for output: default is itemid. Just \ gives blank heading.										
4-6	Not used/reserved.										
7	V/CONV (input/output conversion). Code(s) separated by value marks (CTRL+]) to process data after sorting or selection, but before output (multiple codes processed from left to right). Also to process values specified in an English command before comparison with attribute's values ('input' processing, usually inverse of output, where defined).										
8	V/CORR (pre-processor). Code(s) separated by value marks (CTRL+]) to process data before sorting or selection and before output conversion applied.										
9	Layout of columnar output: <table border="1"> <tr> <td>I</td><td>left justified with wrapped text indented one space;</td></tr> <tr> <td>L</td><td>left justified with value folded if it exceeds column width;</td></tr> <tr> <td>R</td><td>right justified with value overlaying previous column if it exceeds column width;</td></tr> <tr> <td>T</td><td>left justified with text folded at blanks;</td></tr> <tr> <td>U</td><td>left justified with value overriding specified column widths so as to print on one line.</td></tr> </table>	I	left justified with wrapped text indented one space;	L	left justified with value folded if it exceeds column width;	R	right justified with value overlaying previous column if it exceeds column width;	T	left justified with text folded at blanks;	U	left justified with value overriding specified column widths so as to print on one line.
I	left justified with wrapped text indented one space;										
L	left justified with value folded if it exceeds column width;										
R	right justified with value overlaying previous column if it exceeds column width;										
T	left justified with text folded at blanks;										
U	left justified with value overriding specified column widths so as to print on one line.										
10	Column width in characters. Overridden by column heading (see attribute 3), if larger.										



About NEC Software Solutions

Our customers change lives, so we create software and services that get them better outcomes. By innovating when it matters most, we help to keep people safer, healthier and better connected worldwide.

NECSWS.com

1st Floor, Bizspace, IMex Centre,
575-599 Maxted Rd,
Hemel Hempstead HP2 7DX
+44 (0)1442 768445

NEC