SMA STANDARD

SMA/Dictionary and Data Structure Specification

SMA: 301

March 1987

SPECTRUM MANUFACTURERS ASSOCIATION

sma

-- NOTICE --

SMA standards are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining, with minimum delay, the proper product for his particular need.

Some material contained herein is designated as proprietary by individual member companies of SMA listed below. Any unauthorized use of such proprietary information is prohibited.

Published by
SPECTRUM MANUFACTURERS ASSOCIATION
9740 Appaloosa Rd., Suite 104
San Diego, CA 92131

Foreword: This document provides a set of definitions for the SMA/Dictionary And Data structure. This structure, provided by all Spectrum Manufacturing Association members, is the foundation for representation of all information within SMA systems. The structure uses a set of definition items to define the various accounts and data files within the system. This document serves as a guide to the preparation of SMA/Dictionary And Data Definition Items that can be moved from one SMA system to another. For details on any specific system, the user should refer to the manufacturer's reference manual.

The SMA Executive Board wishes to thank the following individuales and organzations for their contributions to the preparation of this document:

D. Harman,	Systems Management, Inc.
C. Saunders,	Fujitsu Microsystems of America, Inc.
K. Hoppe,	Altos Computer Systems
I. Sanders,	CIE Systems
H. Eggers,	Mcdonnell Douglas Computer Systems Co.
C. Wilson,	General Automation, Inc.

CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

1.0 Scope

1.1 Specification Objectives:

It is the objective of this document to provide the user with a definition of the SMA/Dictionary And Data Structure to enable the design of data files and the preparation of Data Definition Items that can be moved from one system to another with maximum portability.

1.2 Inclusions:

This document includes a definition of the SMA/Data Base system, File structure, Dictionaries and Data Definition Items, used by all the SMA systems to define the Data Base.

1.3 Exclusions:

Extensions to the Dictionary or Data structures may have been implemented by individual SMA manufacturers. This document excludes any such items which have not been adopted by a majority of the voting SMA member companies. Also, this document does not address "controlling-dependent" relationships, which are reserved for a later edition of the standard. This document does not include data structure information related to the WITHIN connective. Although many systems support "user exits", they are not considered to be within the scope of the SMA Standards.

2.0 Definitions

2.1 Names and Symbols:

The names for verbs and reserved words used in this document are used by all the SMA manufacturers. They can be easily changed for different languages without affecting the documented functionality.

The use of quotes (") and single quotes (') is required in the forms where shown.

The use of braces ({ }) means that the included string is optional.

The use of ellipsis (...) means that the preceding information can be repeated.

2.2 Structural Terms:

Item-Id - A character string (maximum 48 characters) which uniquely identifies an item within a file. The item-id may include any characters except system delimiters, however the use of blanks, single quotes, quotes, commas, and backslash characters may require special considerations.

Item Body - The variable length character structure which makes up the information content of the item. It is composed of any number of attributes, values, and subvalues. A segment mark is used to mark the end of the structure.

IDP - Indirect Data Pointer: Special type of item body that locates the data stored separately from the item body. Used to store non-character structures such as programs, as well as other extended structures.

SM - Segment Mark: Delimiter character used to mark the end of an item body. It is represented graphically by an underscore. A segment mark may not be included within data.

AM - Attribute Mark: Delimiter character used to mark the end of an attribute. It is represented graphically by an up-arrow. The last AM in an item is implied by the presence of a segment mark.

VM - Value Mark: Delimiter character used to mark the end of a value. It is represented graphically by a right bracket. The last VM in an attribute is implied by an attribute or segment mark.

2.2 Structural Terms (continued):

SVM - Subvalue Mark: Delimiter character used to mark the end of a subvalue. It is represented graphically by a backslash. The last SVM in a value is implied by a value, attribute, or segment mark.

BM - Buffer Mark: Delimiter character used to mark the beginning or ending of special character strings. It is represented graphically by a left bracket.

AMC - Attribute Mark Count: The positional count, from 1, that locates a specific attribute within the body of an item. An AMC value of zero is used to locate the item-id.

VMC - Value Mark Count: The positional count, from 1, that locates a specific value within a multivalued attribute.

SVMC - Subvalue Mark Count: The positional count, from 1 that locates a specific subvalue within a multisubvalued structure.

3.0 Dictionary And Data Structure

3.1 System File Structure:

The SMA systems use the SMA-based file pointer system which is a hierarchical structure consisting of up to four levels.

The top level, known as level zero, is the System Dictionary file and contains the File Definition Items for the Master Dictionary file of each of the Accounts on the system. This file also contains the File Definition Items for system wide files.

Level one, known as the Master Dictionary (MD) level file for an account, contains File Definition Items, File Synonym Items, Data Definition Items, stored procedures, verbs, and vocabulary words for the SMA/Retrieval language.

Level two, known as a Dictionary Level file, contains the File Definition Items for data portions of the file and optionally Data Definition Items.

Level three, known as the data level file, contains data items. File Definition Items and File Synonym Items, if present, will be ignored if accessed at the data level.

3.2 File Definition Item:

The File Definition Item is stored in the dictionary that is associated with the file.

The File Definition Item has the following form.

AMC	Description
Item-Id	The name by which the file is referenced.
1	Defines the item to be a pointer to a file. It must contain one of the following forms:
	D — The file is of standard form containing items that are saved on a file save operation.
	DX — The file is the same as a D file, except that the file will not be saved on a file save operation.

3.2 File Definition Item (continued):

AMC	Description
1	Forms of File Definition Item type, continued:
	DY The file is the same as a D file, except that only valid File Definition Items will be saved on a file save operation; other types of items will not be saved. Note: File Definition Items are only valid in file levels Ø, 1, and 2. When the file save operation encounters any valid File Definition Item, it proceeds to save the indicated file.
	DC The file is the same as a D file, but the file may also contain indirect data pointer items that are saved on a file save operation.
	DCX The file is the same as a DC file, except that the file will not be saved on a file save operation.
	DCY The file is the same as a DC file, except that only valid File Definition Items will be saved on a file save operation; other types of data will not be saved. Note: File Definition Items are only valid in file levels Ø, 1, and 2. When the file save operation encounters any valid File Definition Item, it proceeds to save the indicated file.
2	Base . . . . | These three attributes define the physical file structure and must not be modified by the user.
3	Modulo . . . |
4	Separation . |
5	The access lock codes used for access protection, represented in a multi-valued list.
6	The update lock codes used for update protection, represented in a multi-valued list.

3.2 File Definition Item (continued):

AMC	Description
7	External processing codes used to convert the item-id between the processing format and the external format. Multiple processing codes can be used and are separated by value marks.
8	Reserved.
9	Defines the justification of data in the output form of the element. This attribute, if included, must be an "L", "R", or "U". The code is used in formatting the output, and in determining the sort sequence when sorting the data and is: L,U Specifies a left-to-right sort, and will left-justify, without folding. This may cause the field to be overlayed by the next attribute. R Specifies a right-justified numeric sort (including alphanumeric elements) which may overlay the previous attribute. Default if not specified is L. File Definition Items for level one files may contain combinations and/or additional codes. These codes are used for other system processors and their meaning is specified elsewhere. If multiple codes are present, then only the first code is used for SMA/Retrieval purposes.
10	Defines the maximum column width when displaying the item-id. The modifier ID-SUPP may be used to suppress the output of the item-id. Default value is 9.
11	Reserved.
12	Reserved.
13	Reallocation parameter which is of the form: (new.modulo,new.separation) where the new.modulo and new.separation will be used for attributes 3 and 4 upon the file create as a part of a file restore.

Note that AMC 5-13 are optional.

3.3 File Synonym Item:

A file may be referenced by the use of a file synonym. Multiple file synonyms can exist for the same data file and may be stored in the three dictionary levels.

The File Synonym Item has the following form:

AMC	Description
Item-id	The name by which the file is referenced.
1	The 'Q' indicates that this is a File Synonym Item.
2	The name of the account in which the the file has been defined. See table below.
3	The name of File Definition Item of the file defined. See table following AMC 10.
4	Reserved.
5	Reserved.
6	Reserved.
7	External processing codes used to convert the item-id between the processing format and the external format. Multiple processing codes can be used and are separated by value marks.
8	Reserved.

3.3 File Synonym Item (continued):

AMC	Description
9	Defines the justification of data in the element. This attribute, if included, must be an "L", "R",or "U". The code is used in formatting the output, and in determining the sort sequence when sorting the data and is: L,U Specifies a left-to-right sort, and will left-justify, without folding. This may cause the field to be overlayed by the next attribute; R Specifies a right-justified numeric sort (including alphanumeric elements) which may overlay the previous attribute. Default if not specified is L. File Synonym Items for level one files may contain combinations and/or additional codes. These codes are used for other system processors and their meaning is specified elsewhere. If multiple codes are present, then only the first code is used for SMA/Retrieval purposes.
10	Defines the maximum column width when displaying the item-id. The modifier ID-SUPP may be used to suppress the output of the item-id. Default value is 9.

Note that AMC 2-10 are optional.

3.3 File Synonym Item (continued):

Reference table for attributes 2 and 3 in File Synonym Items:

AMC-2	AMC-3	Description
account	file	Form used to reference a file in a specified account. It may be this account or some other account.
null	file	Form used to reference a file in this account by another name.
null	null	Form used to reference the dictionary of this file without the use of DICT.
account	null	Form used to reference the master dictionary of another account.

If attributes 7, 9, and 10 exist in the file synonym definition, they take precedence over those attributes in the File Definition Item when referencing the file via the file Synonym Item.

3.4 Data Definition Item:

Data Definition Items are used to define the data structure of the associated data file. The item-ids of these items are used in the SMA/Retrieval Language sentences in selection criteria, sort criteria, and output criteria.

Data Definition Items have the following structure:

AMC	Description
Item-Id	The name by which this data definition is referenced.
1	An 'A', 'S', or 'X' identifies the item as a Data Definition Item. Note that 'A' and 'S' have identical functionality but may be used by the application designer to identify primary 'A' or synonym 'S' attribute data definitions. 'X' is used as place holder for a Data Definition Item in numeric default group of Data Definition Items. The value of an attribute with a Data Definition Item containing an 'X' will not be output. Attribute Data Definition Items with a 'X' should never be explicitly named in a SMA/Retrieval language sentence.
2	The numeric value (AMC) locating the attribute in the item which is being defined.
3	Textual data used as a column heading in LIST or SORT sentences. If null, the item-id is used as the heading. May contain blanks for formatting. The reserved character "\" is used to specify a null heading. Multiple line headings for columnar listings may be specified by storing multiple values.
4	Defines the 'controlling-dependent' relationship.
5	Reserved.
6	Reserved.
7	External processing codes used to convert between the processing format and the external format. Multiple processing codes can be used and are separated by value marks.

3.4 Data Definition Item (continued):

AMC	Description
8	Internal processing codes used to convert from internal format to processing format. Multiple processing codes can be used and are separated by value marks.
9	Defines the justification of data in the element. This is a required attribute, and must be an "L", "R", "T", or "U". The code is used in formatting the output, and in determining the sort sequence when sorting the data and is: L Specifies a left-to-right sort, and will left-justify, folding long strings at the end of the column width defined by attribute 10; R Specifies a right-justified numeric sort (including alphanumeric elements); T Specifies a left-to-right sort, and will left-justify, folding long strings at blanks; U Specifies a left-to-right sort, and will left-justify, without folding. Default if not specified is L.
10	Defines the maximum column width when displaying the attribute. A value of zero may be used to suppress output on detail lines. The default value is 9.

Note that AMC 2-10 are optional.

3.5 Item Structure:

An item is said to be made up attributes, each of which may be made up of values, each of which may be made up of subvalues.

An element refers to the data in an attribute, value, or subvalue. It may be null, a numeric string, or a character string.

3.5.1 Item-Id:

Each item within a file has associated with it a unique item-id. This item-id may be referenced as attribute Ø within Data Definition Items and processing codes.

3.5.2 Attribute:

An attribute is a data element within an item. Attributes are sequentially numbered starting from one and are delimited by attribute marks. A given attribute typically contains data with the same context in all the items in a particular file. The SMA/Retrieval language assumes that all items participating in a particular sentence/report contain the same context of data within any given attribute.

3.5.3 Multi-Valued Attributes:

An attribute containing one or more value marks is said to contain multi-values, sequentially numbered starting from one.

3.5.4 Multi-Valued Values:

A value containing one or more subvalue marks is said to contain multi-subvalues, with the subvalues sequentially numbered starting from one.

4.0 Processing Codes

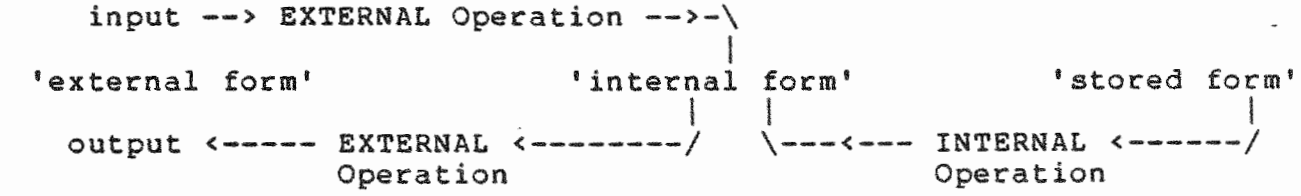
The SMA/Dictionary and Data Structure provides a set of processing codes that can be specified in attribute 7, where they perform EXTERNAL operations, or in attribute 8, where they perform INTERNAL operations.

During the processing of a SMA/Retrieval language sentence, the data in the items can exist in three different forms. The first is the 'stored' form. Whenever the element is retrieved from an item it is picked up in the stored form.

The INTERNAL operations, if specified, are applied to the element, converting it from the 'stored' form to the 'internal' form. The 'internal' form of the Data Definition Item is used whenever it:

1. is part of a sort.criteria,
2. is compared to a selection.criteria,
3. is compared for an output limiter,
4. is used for a TOTAL or GRAND-TOTAL computation,
5. produces a control break,
6. is printed, except on break lines,
7. is output by reformat or select output.criteria.

The EXTERNAL operations, if specified, are applied to convert an element between 'internal' form and 'external' form. The 'external' form is used for user input and output. If a Data Definition Item contains an EXTERNAL operation and it is followed by selection.criteria in a SMA/Retrieval Language sentence, then the EXTERNAL operation is applied as an input conversion. The select.criteria value is converted from 'external' form to 'internal' form. Print.limiters and explosion.limiters are converted in a similiar manner. This action causes selection, sorting, and limiting to always operate on the 'internal' form of the data elements.



4.1 Processing Codes Summary:

CODE	DESCRIPTION
A	ALGEBRIAC. Used to evaluate algebriac expressions.
C	CONCATENATE. Used to concatenate elements.
D	DATE. Used to convert dates.
F	FUNCTION. Used to manipulate elements.
G	GROUP. Used to extract one or more fields separated by a given non-system delimiter.
L	LENGTH. Used to validate the length of an element.
MC	MASK CHARACTER. Used to convert strings to upper or lower case, or to extract alphabetic or numeric characters from strings.
ML	MASK DECIMAL. Used to format and scale numbers, left justified.
MR	MASK DECIMAL. Used to format and scale numbers, right justified.
MT	MASK TIME. Used to convert time.
MX	MASK HEXADECIMAL EXPANSION. Used to convert ASCII characters to their hexadecimal representations.
MY	MASK HEXADECIMAL COMPRESSION. Used to convert hexadecimal characters to their ASCII representation.
P	PATTERN MATCH. Used to validate elements against a specified pattern.
R	RANGE. Used to validate elements which fall within the specified numeric ranges.
S	SUBSTITUTION. Used to generate alternative data for the referenced element.
T	TEXT EXTRACTION. Used to extract a fixed field from an element.
Tfile	FILE TRANSLATION. Uses the element as an item-id to access attributes in the specified file.

4.2 Algebriac Processor, A CODE:

The 'A' code provides functions similar to the 'F' code, and is written in an algebriac form using infix notation. For example, A1+2 would add the element from attribute 1 to the element of attribute 2. Evaluation of the expression proceeds from left to right unless reordered by use of parentheses. The infix operators have no order of precedence defined. The form is:

A expression {operator expression}...

where:

expression = operand {operator operand}

Parenthesis may be used as required to control the order of expression evaluation. The inner-most parenthesis expression will be evaluated first. The term "expression-1" is used to refer to the operand or expression on the left side of an operator, and the term "expression-2" to the operand or expression on the right side.

The permissible operands are:

amc{R{R}} An Attribute Mark Count specifies the number of the attribute from which the element is to be retrieved for use in the operation. If the AMC is followed by R, it specifies that the first value of an attribute is to be used repeatedly when evaluating with other multi-valued attributes. If a second R is present, the first subvalue is used repeatedly for evaluation with other multi-subvalued attributes.

N(name) The character N followed by the name of a Data Definition Item enclosed in parentheses can be used to specify the element to be used in the operation. The 'name' must exist in the dictionary being used. The data element retrieved will be specified via attribute 2 unless the Data Definition Item referenced contains a function. If this occurs, the function will be performed.

literal A alpha-numeric literal string is specified by being enclosed in either single quotes or double quotes.

4.2 Algebriac Processor, A Code (continued):

The following special system counters and values may be used as operands in an A-code expression. Refer to the section on F-code for further details. They are:

NI	the item counter
NV	the value counter
NS	the subvalue counter
ND	the detail line counter
NB	the break level counter
D	the system date (in internal format)
T	the system time (in internal format)

There are several special functions which may be used as operands in the A-code expression. They are:

R(exp,exp)	The Remainder function takes two expressions as operands, and returns the remainder of the first operand divided by the second.
S(exp)	The Summation function computes the summed total of the enclosed expression for all elements of a multivalued or multisubvalued set.
expl[exp2,exp3]	A sub-string of an element is specified. Expl is the element from which the substring is to be extracted, exp2 is the character in the element from which to start the extraction, and exp3 is the number of characters to be extracted.

4.2 Algebriac Processor, A Code (continued):

The arithmetic and relational operators all require two operands. The arithmetic operators are:

+	adds the two expressions and returns the sum.
-	subtracts expression-2 from expression-1 and returns the difference.
*	multiplies the two expressions and returns the product.
/	divides expression-1 by expression-2 and returns the quotient. The quotient is always an integer but may have implied scaling or a decimal point.
:	concatenates the second operand onto the end of the first operand. The operands are considered as character strings and the result is a character string.

The relational operators return a one if the relational operation evaluates as true and a zero if the relational operation evaluates a false. The relational operators are:

=	Expression-1 equal to expression-2.
#	Expression-1 not equal to expression-2.
<	Expression-1 less than expression-2.
>	Expression-1 greater than expression-2.
>=	Expression-1 greater than or equal to expression-2
<=	Expression-1 less than or equal to expression-2.

In the absence of parentheses to indicate the order in which operators are to be applied, operations proceed in straight left to right sequence, with no precedence among operators.

4.3 Concatenation, C Code:

The 'C' code provides the facility to concatenate attributes and/or literals prior to output. The form is:

C;op{x op}...{x}

x — is the character to be inserted between the concatenated attributes and/or literals. A semicolon is a reserved character that means no separation character is to be used. Any non-numeric (except system delimiters) is valid, including a space.

op — is the attribute mark count (AMC); or any string enclosed in single quotes, double quotes, or backslashes; or an asterisk, which specifies the last generated value from a previous operation is to be used.

4.4 Date Conversion, D Code:

The 'D' code provides for the conversion of dates in internal format to external for output or from external format to internal format when used with selection.criteria. December 31, 1967 is defined as day zero with positive values following and negative values earlier. Many of the date conversions will not operate on input data as they do not uniquely define a month, day, and year. The form is:

D{n}{xm}{s}

n — is an option single digit number which specifies the number of digits to occur in the year on output. If 'n' is 0, no year will appear in the the date. The only valid digits for 'n' are 0,1,2,3, or 4. If 'n' is not specified then n = 4 is assumed.

x — Stands for any single non-numeric character which specifies delimiter between fields for group extract. The 'x' cannot be one of the system delimiter.

m — is a single digit number that must accompany 'x' (if 'x' is specified). 'm' specifies the number of fields to skip for group extraction. The group extraction is done before the conversion is performed.

4.4 Date Conversion, D Code (continued):

s is either any non-numeric character that may be specified to separate the day, month, and year on output, or special date sub-code. If 's' is specified, the output format will be MMsDDs{{{{Y}Y}Y}Y}. If 's' is not specified, the date output format will be DD MMM {{{{Y}Y}Y}Y}. (MM is a two digit month, MMM is a three character alpha month abbreviation.) On External to Internal conversion of a two digit year, the years range from 1930 through 2029.

The permissible date sub-codes are:

D Day of the month.

I Internal format, Reverse conversion.

J Julian day of year.

M Month numeric.

MA Month alphabetic.

Q Quarter numeric.

W Weekday numeric (Monday=1, Sunday=7).

WA Weekday alphabetic.

Y Year. Default = 4 digits.

4.5 Function Processor, F Code:

The 'F' code processor uses a 15 element post-fix push-down stack for storing values. An operation specified by an F-code operates on the last one, two or three entries pushed onto the stack. Entries are removed from the stack as they are used in the operation. The results of the operation is pushed onto the stack. This continues for each operator until the entire F-code is processed. The final result is then the value on the top of the stack. The form is:

FS;elm{;elm...}

Note that this form of the 'F' code processor differs from previous implementations in the use of the S to designate standard form. This standard form includes all ordered binary operations in classic reverse Polish ordering, including the comparison operations.

4.5 Function Processor, F Code (continued):

where 'elm' may be any of the following:

amc{R{R}} A numeric Attribute Mark Count specifying the element to be pushed onto the stack. If the AMC is followed by R, it specifies that the first value of an attribute is to be used repeatedly when evaluating with a multi-valued attribute. If the second R is present, it specifies that the first subvalue of a value is to be used repeatedly.

Cn A capital 'C' followed by a string, specifies that the string is to be pushed onto the stack. The string is ended by the next semicolon.

D specifies that the current date is to be pushed onto the stack (internal format).

literal The literal string enclosed in either single or double quotes is pushed onto the top stack entry.

T specifies that the current time is to be pushed onto the stack (internal format).

NA specifies that the number of attributes in the item is to be pushed onto the stack.

NB specifies that the current Break level number is to be pushed on to the stack. 1 = the lowest level break and 255 = the grand-total line.

ND specifies that the number of items since the BREAK on a break line is to be pushed onto the stack. If on a GRAND-TOTAL line, it equals the item count.

NI specifies the value of the current item counter is to be pushed onto the stack (number of items listed or selected).

NL specifies that the length of the item is to be pushed onto the stack.

NS specifies the current subvalue counter, for columnar listing only, is to be pushed onto the stack.

NV specifies the current multi-value counter, for columnar listing only, is to be pushed onto the stack.

4.5 Function Processor, F Code (continued):

where 'elm' may be any of the following:

LPV	specifies the loading of the value from the previous processing code.
*{n}	Multiplication of the top two stack entries. If 'n' is specified, the result is divided by 10 raised to the power of n.
/	Divide the second stack entry by the top stack entry and replace the top stack entry with the quotient.
R	Divide the second stack entry by the top stack entry and replace the top stack entry with the remainder
+	Add the second stack entry to the top stack entry and replace the top stack entry with the sum.
-	Subtract the top stack entry from the second stack entry and replace the top stack entry with the difference.
:	The top stack entry is concatenated onto the end of the second stack entry, and the resulting concatenated string replaces the the top stack entry.
[]	A subset from the third stack entry is extracted, using the second stack entry as the starting character position, and the top stack entry as the number of characters to be extracted; the result is placed in the top stack entry.
S	A total sum of all previous computation is placed on the top of the stack. The sum operator is used with multi-valued or multi-subvalued elements to produce a single value. Multiple S operators may be present within a function. The domain of a function begins at either the start of the function or immediately following the previous S operator. At the conclusion of the S operator, a single value is present on the stack.
_	Exchange the top two stack entries.
P	Duplicates the top stack entry back on to the stack.

4.5 Function Processor, F Code (continued):

where 'elm' may be the following:

(...) A standard conversion operator, enclosed in parentheses, will operate on the top stack entry and the result will replace the original top stack entry.

The following relational 'elm's operate on the top two stack entries, and a result of zero or one is placed in the top stack entry, depending on whether the condition is not or is satisfied.

= Stacks a one if the two top stack entries are equal, and a zero is stacked if they are unequal.

Stacks a one if the two top stack entries are unequal, and stacks a zero if they are equal.

> Stacks a one if the second stack entry is greater than the top stack entry, stack zero otherwise.

< Stacks a one if the second stack entry is less than the top stack entry, a zero otherwise.

] Stack a one if the second stack entry is greater than or equal to the top stack entry, a zero otherwise.

[Stacks a one if the second stack entry is less than or equal to the top stack entry.

4.6 Group Extraction, G Code:

The 'G' code provides the facility to extract from an element one or more contiguous fields separated by a given delimiter. The form is:

G{m}xn

m specifies the number of fields to skip. If m is not specified, zero is assumed, and no fields are skipped.

x represents any single non-numeric character, except any system delimiter, which is the field separator.

n is a decimal number which is the number of contiguous fields to be extracted.

4.7 Length, L Code:

The 'L' code validates the length of an element. The form is:

L{n{,m}}

L	returns the length of the element.
Ln	returns the element if it is less than or equal to 'n' characters long, otherwise a null is returned.
Ln,m	returns the element if it is equal to or greater than 'n' characters long and less than or equal to 'm' characters, otherwise a null is returned.

4.8 Mask Character, MC Code:

The 'MC' code provides the facility to change an element to upper or lower case, to select out certain classes of characters, or convert from hexadecimal to decimal and from decimal to hexadecimal. The forms are:

MCA	Extracts all alphabetic characters from an element.
MC/A	Extracts all non-alphabetic characters from the element.
MCD{X}	Converts decimal element to a hexadecimal element.
MCL	Converts an element to lower case. Will convert all upper-case letters to lower-case; has no effect on lower-case letters or non-alphabetic characters.
MCN	Extracts all numeric characters (0-9) from an element.
MC/N	Extracts all non-numeric characters form an element.
MCT	Converts first letter following a non-alphabetic character to upper-case. The characters up to the next non-alphabetic character are converted to lower-case. This process is repeated through-out the entire element.
MCU	Converts an element to upper case. Will convert all lower-case letters to upper-case; has no effect on upper-case letters or non-alphabetic characters.
MCX{D}	Converts hexadecimal element to a decimal element.

4.9 Mask Decimal, ML And MR Codes:

The 'ML' and 'MR' codes provide the facility to do special output formatting of data elements. The ML code specifies that the output is to be left justified and the MR code specifies that the output is to be right justified. The forms are:

```
ML{n{m}}{Z}{,}{cr}{$}{(format-string)}
MR{n{m}}{Z}{,}{cr}{$}{(format-string)}
```

n is a single decimal digit (0-9) which specifies the number of digits to be printed to the right of the decimal point. If 'n' is not specified, 0 assumed. If 0 is assumed or specified, no decimal point is printed.

m is a single digit numeric (0-9) which specifies that the element is to be divided by that power of ten. The number 'm' is the number of implied digits to the right of the decimal point. If m > n, then the element will be rounded to 'n' digits.

Z specifies zero-suppression. An element of 0 (zero) will be printed as blanks.

, specifies the insertion of a comma every three digits to the left of the decimal point.

cr specifies the designation of debit/credit symbols as:

- C causes negative elements to be followed by the letters CR.
- D causes positive elements to be followed by the letters DB.
- E causes negative elements to be enclosed inside angle brackets.
- M causes negative elements to be followed by a minus sign.
- N causes the minus sign on negative elements to be suppressed.

In the absence of a credit symbol, negative numbers are presented with a leading minus sign.

$ causes a currency symbol to be appended to the front of the element before justification.

4.9 Mask Decimal, ML And MR Codes (continued):

The format mask specification, which is enclosed in parentheses, consists of format codes and literal data. The format codes are one of the characters #, *, or %, optionally followed by a number to specify that number of repetitions of the characters. The meaning of the format codes are:

#{n}	specifies that the element is to be justified in a field of 'n' blanks.
*{n}	specifies that the element is to be justified in a field of 'n' asterisks.
%{n}	specifies that the element is to be justified in a field of 'n' zeroes.

NOTE: Any other character, including parentheses may be used as a field fill. Mixed mode fields may be formed by repeating the control characters (#, *, and %).

4.10 Mask Time Conversion, MT Code:

The 'MT' code provides the facility for converting times to or from internal format. The internal time format is the number of seconds. The form is:

MT{H}{S}

H	is the capital letter H, which specifies 12 hour format. If 'H' is omitted, 24 hour (international) format is assumed.
S	is the capital letter S, which specifies seconds on on output. If 'S' is omitted seconds are not listed on output.

When the codes MTH or MTHS are used, AM or PM is always displayed immediately following the 12 hour time.

4.11 Mask Hexadecimal Expansion, MX Code:

The 'MX' code provides the facility to convert an element, one byte at a time, into the corresponding hexadecimal representation. Each character will be converted to a 2-byte hexadecimal number. On input conversion, the element will be considered right justified. The form is:

MX

4.12 Mask Hexadecimal Compression, MY Code:

The 'MY' code provides the facility to convert a hex element, two bytes at a time, into the corresponding ASCII representation. Each character will be converted to a 1-byte ASCII character. The form is:

MY

4.13 Pattern Matching, P Code:

The 'P' code validates an element if it matches any of the specified patterns. If the element does not match any pattern, a null is returned. The form is:

P(op){;(op)}...

Any combination of the following forms is valid within an 'op'.

op	Description
nN	The integer number 'n' followed by the letter 'N', tests for n numeric characters.
nA	The integer number 'n' followed by the letter 'A', tests for n alpha characters.
nX	The integer number 'n' followed by the letter 'X', tests for n characters.
'literal'	A literal, enclosed in single quotes ('), tests for the literal string.

4.14 Range, R Code:

The 'R' code validates an element which falls within a specified range. The form is:

Rn,m{;n,m}...

n is the starting integer of the range.

m is the ending integer of the range.

If the range specifications are not met, null is returned.

4.15 Substitution, S Code:

The 'S' code substitues the element of the reference attribute with the element of the first specified attribute or a literal if the element of the reference attribute is not null or zero. If the element of the reference attribute is null or zero, then it will be substitued with the element of the second specified attribute or a literal. The form is:

S;op1;op2

op1 if the element is not null or zero, then this attribute or literal (enclosed in single quotes) is used for substitution.

op2 if the element is null or zero, then this attribute or literal (enclosed in single quotes) is used for substitution.

Note that an asterisk, which specifies the last generated value from a previous operation, may be used as either 'op1' or 'op2'.

4.16 Text Extraction, T Code:

The 'T' code extracts a contiguous string of characters from an element. The form is:

T{m,}n

m is the optional starting column number.

n is the number of characters to be extracted.

The form 'Tm,n' counts columns and extracts characters from left to right of the element, regardless of the code in attribute 9 of the Data Definition Item.

The form 'Tn' extracts 'n' characters either from the left or the right, depending upon on the code in attribute 9 of the Data Definition Item. If that code is not 'R', then the 'Tn' form extracts the first 'n' characters of the element. If that code is 'R', then the 'Tn' form extracts the 'n' rightmost characters of the element.

4.17 File Translation, Tfile Code:

The 'Tfile' code provides the facility for converting an element by translating through a file. The element to be translated is used as an item-id for retrieving an item from the specified translation file. The translated element is retrieved from the specified attribute of the item. The form is:

T{DICT }file;c{n};{i-amc};{o-amc{;b-amc}}

DICT specifies the use of the dictionary of the file instead of the data portion of the file.

file specifies the name of the file to be used for the translation.

c is the translation sub-code, which is one of the following:

- V Conversion item must exist on file, and the specified attribute must have an element. Aborts with an error message if translation is impossible.
- C Convert if possible; use original element if item in translate file does not exist or has a null conversion attribute.
- 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.
- X Convert if possible, otherwise return a null element.

n is a value mark count. If 'n' is specified, then only the element in value n will be returned. If 'n' is not specified, then all the values in the element are concatenated together with blank separators and returned.

i-amc is the decimal attribute number for input translation. If the i-amc is omitted, no input translation takes place.

o-amc is the numeric attribute number for output translation.

b-amc if specified, will be used instead of o-amc during the listing of break-on and total lines.

THIS IS THE LAST PAGE