

NAKED MINI® 4

OBJECT PROGRAM FORMAT

21127-90



Engineering Reference Number
90-21127-03D0

NAKED MINI Division

HERTFORD HOUSE
DENHAM WAY, MAPLE CROSS
RICKMANSWORTH, HERTFORDSHIRE WD3 2XD

Telephone Rickmansworth 71211

Telex 922654

© 1979 Computer Automation, Inc.

REVISION HISTORY

<u>Revision</u>	<u>Issue</u>	<u>Comments</u>
A0	Dec 1978	Original issue
B0-C0		Skipped for configuration control.
D0	Aug 1979	General Revision; to be used with CS4-D0 and RTX4/IOS4-D0 revisions or later. The information in the manual corresponds to software released at revision level D1.

Computer Automation, Inc., reserves the right to make changes in specifications at any time and without notice. The information furnished by Computer Automation, Inc., in this publication is believed to be accurate and reliable. However, no responsibility is assumed by Computer Automation, Inc., for its use; nor for any infringements of patents or other rights of third parties resulting from its use. No license is granted under any patents or patent rights of Computer Automation, Inc.

Computer Automation, Inc., welcomes comments and suggestions for improvements to our manuals. Should difficulty be encountered with the information in the manual or if you have a suggestion for manual improvement, please send the comments or suggestions to Computer Automation, Inc., Technical Publications, Department 11-0715, 18651 Von Karman, Irvine, California, 92713. Please provide as much information as possible to help evaluate the comment or suggestion.

This Programming Note describes the NAKED MINI® 4 representation of object programs on external media. This information is provided for programmers who wish to create generators (i.e., software programs which produce object programs) and/or processors (i.e., software programs which read and use object programs) to interact with Computer Automation software. The Object Program Format describes TC (Type Code) Records, TC Record layouts, and type code functions. Type codes are listed and defined in numerical order.

TC (TYPE CODE) RECORDS

A recorded object program is composed of a series of TC (Type Code) Records. TC Records are successive bytes of related data. Each object program record can contain any number of TC Records. On NAKED MINI® 4 Family computers, a single TC Record may cross an object program record boundary.

A TC Record always starts with a one-byte control field called the type code. Most type codes signal the beginning of a TC Record of some fixed length and layout; a few type codes require a Count field to indicate the exact length of the associated TC Record. No special marker is needed at the end of a TC Record; the next type code follows immediately.

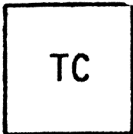
TC RECORD LAYOUTS

All NAKED MINI® 4 TC Records are in one of 13 standard layouts, presented on the next pages. These layouts are built from a few basic elements, each with a fixed length and functions. These elements are identified in the layout presentations as follows:

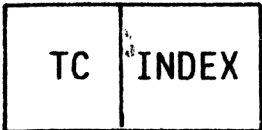
BYTES	One byte indicating the number of bytes in the layout
COUNT 16	Two bytes which contain a positive binary number, having a value between :0000 and :FFFF, or 0 and 65535 decimal.
COUNT 8	One byte containing a positive binary number, with a value between :00 and :FF, or 0 and 255 decimal.
FLAGS	One byte of the bits used to identify the attributes of a given segment.
INDEX	One byte representing an index (NAM, EXTERNAL, CHAIN, TASK NAME (TN) or SEGMENT) to a symbol list.
MASK	Two bytes defining the field within the skeleton to be affected by the operation.
SHIFT	One byte indicating the number of places to shift.

The type code name, function, record layout, and order in program unit can be found in the table, NAKED MINI® 4 Type Codes.

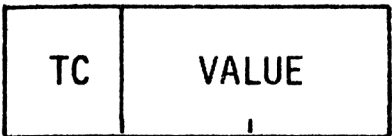
Layout A (one byte):



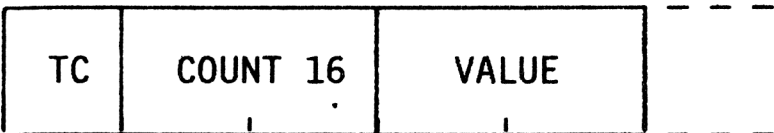
Layout B (two bytes):



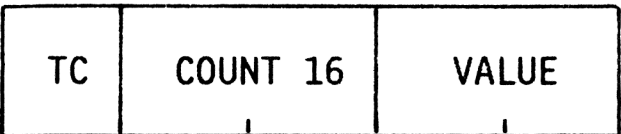
Layout C (three bytes):



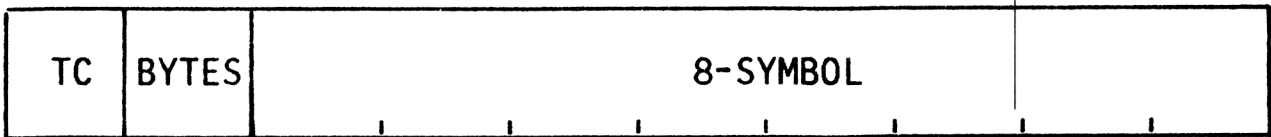
Layout D (COUNT 16*2+3 bytes):



Layout E (five bytes):



Layout F (BYTES+2 bytes):



NAKED MINI 4 OBJECT PROGRAM FORMAT

90-21127-03

Layout G (six bytes):

TC	INDEX	VALUE	MASK
----	-------	-------	------

Layout H (BYTES+2 bytes):

TC	BYTES	VALUE	FLAGS
----	-------	-------	-------

Layout J (BYTES+2 bytes):

TC	BYTES	Undefined
----	-------	-----------

Layout K (four bytes):

TC	SHIFT	MASK
----	-------	------

Layout L (two bytes):

TC	COUNT 8
----	---------

Layout M (three bytes):

TC	INDEX	COUNT 8
----	-------	---------

Layout N (five bytes):

TC	TN	INDEX	VALUE
----	----	-------	-------

Table 1. NAKED MINI® 4 Type Codes

TYPE CODE	RECORD LAYOUT	ORDER IN PROG. UNIT	NAME	FUNCTION
:02 ¹	C	7	TCENDA	End Program, Absolute
:03 ¹	C	7	TCENDR	End Program, Relocatable
:04 ¹	C	6	TCORGA	Origin, Absolute
:05 ¹	C	6	TCORGR	Origin, Relocatable
:06 ¹	D	6	TCTAB	Data, Absolute
:07 ¹	D	6	TCTRLW	Data, Relocatable
:08 ¹	E	6	TCRESF	Reserve and fill here
:09 ¹	C	6	TCBRLW	Byte Address Constant
:18 ¹	C	1	TCBGN3	Begin Program
:40 ²	A	6	TCNULL	Null
:46	C	6	TCTMPRL	Reserved
:47	N	6	TCINTY	Initial Y-register
:48	B	6	TCTMPL	Template Address
:49	F	3	TCTASK	List Task Names
:4A	F	2	TCENTP	List primary definitions
:4B	F	4	TCEXTP	List primary externals
:4C	F	4	TCENTS	List secondary definitions
:4D	F	4	TCEXTS	List secondary externals
:4E	F	4	TCCHNP	List chains
:4F ²	F	4	TCSGMT	List segment names
:56	A	6	TCMNEG	Stack negate (unary) operator
:57	A	6	TCMSFT	Stack shift operator
:58	A	6	TCMAND	Stack AND operator
:59	A	6	TCMIOR	Stack Inclusive OR operator
:5A	A	6	TCMXOR	Stack Exclusive OR operator
:5B	A	6	TCMNOT	Stack logical NOT (unary) operator
:5C	A	6	TCMEQ	Stack relational equate operator
:5D	A	6	TCMNE	Stack relational not equal operator
:5E	A	6	TCMGT	Stack relational greater than operator
:5F	A	6	TCMLT	Stack relational less than operator
:60	A	6	TCMGE	Stack relational greater than or equal operator
:61	A	6	TCMLE	Stack relational less than or equal operator
:62	K	6	TCKOFD	Field from stack (stack terminator)
:63	G	6	TCINTA	Reserved
:64	G	6	TCINTR	Intermediate access, relocatable
:66	A	6	TCZOP	Stack definition
:67	B	6	TCDENP	Primary definition
:68	B	6	TCDENS	Secondary definition
:69	B	5 & 6	TCLOC	Location counter
:6A	A	6	TCKOP	Begin stack operation
:6B	B	6	TCNCHP	Chain element
:6C	B	6	TCHCHP	Chain head
:6D	J	6	TCNOTE	Note record
:6F ^{2, 3}	H	4	TCSGML	List segment attributes

¹Processed by AutoLoad. ²Ignored by AutoLoad.

³The TCSGMT for a given segment must precede the corresponding TCSGML for that segment.

Table 1. NAKED MINI@ 4 Type Codes (Continued)

TYPE CODE	RECORD LAYOUT	ORDER IN PROG. UNIT	NAME	FUNCTION
:70	A	6	TCMADD	Stack ADD operator
:71	A	6	TCMSUB	Stack SUBtract operator
:72	A	6	TCMMPY	Stack MULTiply operator
:73	A	6	TCMDVD	Stack DIVide operator
:74	B	6	TCTEXP	Text external (primary)
:75	B	6	TCTEXS	Text external (secondary)
:77	L	4	TCNTSK	Reserved
:78	M	6	TCDADR	Reserved

TYPE CODE FUNCTIONS

Type Code :02 Name TCENDA Layout C

Meaning: End absolute program unit

The VALUE field is either a starting address of the source program or a minus one (:FFFF) to indicate the omission of the transfer address. Type code :02 indicates the end of an absolute program unit.

Type Code :03 Name TCENDR Layout C

Meaning: End relocatable program unit

The VALUE field is either a starting address of the source program or a minus one (:FFFF) to indicate the omission of the transfer address. Type code :03 indicates the end of a relocatable program unit. The transfer address is assumed to be relocatable in the segment defined by the last :69 type code.

Type Code :04 Name TCORGA Layout C

Meaning: Absolute origin

The location counter is set to the VALUE field. The absolute value is used.

Type Code :05 Name TCORGR Layout C

Meaning: Relocatable origin

The location counter is set to the VALUE field, relocated by the base address of the segment specified in the last :69 type code.

Type Code :06

Name TCTAB

Layout D

Meaning: Absolute data

The COUNT field indicates how many VALUE fields follow. The VALUE fields are taken as a block of absolute data. If type code :06 is placed in a legal stack expression (i.e., a :66-:67 or :68 sequence or a :6A-:62 sequence), the data generated is placed on the linker reverse polish stack. In any other context this type code causes the data to be deposited at the current value of the location counter and the location counter to be incremented.

Type Code :07

Name TCTRLW

Layout D

Meaning: Relocatable data

The COUNT field indicates how many VALUE fields follow. The VALUE fields are taken as a block of relocatable data. If this type code is placed in a legal stack expression (i.e., a :66-:67 or :68 sequence or a :6A-:62 sequence), the data generated is placed on the linker reverse polish stack. In any other context type code :07 causes the data to be deposited at the current value of the location counter and the location counter to be incremented. The data is relocated by the base address of the segment specified in the last :69 type code.

Type Code :08

Name TCRESF

Layout E

Meaning: Reserve and fill here

Type code :08 causes a block of absolute data (with value VALUE and length count) to be deposited at the current position of the location counter. The location counter is incremented by COUNT.

Type Code :09

Name TCBRLW

Layout C

Meaning: Byte address constant

The VALUE field is taken to be a byte address constant and $VALUE+R+R$ is deposited at the current position of the location counter, where R is the base of relocation of the segment defined by the last :69 type code. The location counter is incremented by one.

Type Code :18

Name TCBGN3

Layout C

Meaning: Begin program

The VALUE field is currently unused and should be zero. This type code must be the first type code of an NM4 object program unit.

Type Code :40

Name TCNULL

Layout A

Meaning: Null operation

Type code :40 is usually used for filler and is ignored.

Type Code: 46

Name TCTMPRL

Layout C

Meaning: Template relative relocatable address (Reserved)

Type Code :47

Name TCINTY

Layout N

Meaning: Initial Y-register

The TN (Task Name) index in this type code refers to the Task Name list generated by the :49 type code, TCTASK, List Task Name. The SEGMENT Index refers to the Segment list generated by the :4F type code, TCSGMT, List Segment Names. During R:BEGIN, the Y-register will be set to the VALUE + the offset of Segment in the expanded template.

Type Code :48

Name TCTMPL

Layout B

Meaning: Template Address

The Index in the type code refers to the Task Name list generated by the :49 type code, TCTASK List Task Names. A word is generated which is the address of the template for the Indexed Task.

Type Code :49

Name TCTASK

Layout F

Meaning: List Task Names

A list is generated as for type code :4A TCENTP, List External (primary) Definitions. This list however is of Task Names and is referenced by type codes :48 TCTASK, Template Address and :47 TCINTY, Initial Y-register. NAM's in a program with type code :49 will be defined only for those tasks listed with one exception: NAM's with the same spelling as a Task Name will be defined for all tasks.

Type Code :4A

Name TCENTP

Layout F

Meaning: List primary definitions

The BYTES field indicates the number of bytes of data which follow. It must be a multiple of 8. The 8-SYMBOL fields which follow are 8-byte symbols, filled with blanks if necessary. A maximum of 32 symbols may be defined in one type code. To define more than 32 symbols, two or more type codes must be used. Multiple :4A type codes must be contiguous. A block of these type codes generates an ordered list of primary definition symbols which are to be referenced later by its INDEX. The first symbol is taken to be INDEX 0, the second INDEX 1, etc. The symbols are defined by a :67 type code.

Type Code :4B

Name TCEXTP

Layout F

Meaning: List primary externals

A list is generated as for type code :4A, except that the list is of primary external symbol references. External symbols are referenced by :74 type codes.

Type Code :4C

Name TCENTS

Layout F

Meaning: List secondary definitions

A list is generated as for type code :4A, except that the list is of secondary external symbol definitions defined by a :68 type code.

Type Code :4D

Name TCEXTS

Layout F

Meaning: List secondary externals

A list is generated as for type code :4A except that the list is of secondary external symbols referenced by :75 type codes.

Type Code :4E

Name TCCHNP

Layout F

Meaning: List Chains

A list is generated as for type code :4A except that the list is of chain symbols referenced by :6B and :6C type codes.

NAKED MINI 4 OBJECT PROGRAM FORMAT

90-21127-03

Type Code :4F

Name TCSGMT

Layout F

Meaning: List segment names

A list is generated as for type code :4A except that the list consists of segment names referenced by type codes :47, :64 and :69.

Stack Type Codes

Names as below

Layout A

<u>Type Code</u>	<u>Name</u>	<u>Meaning</u>
:56	TCMNEG	Stack negate (monadic) operator
:57	TCMSFT	Stack shift operator
:58	TCMAND	Stack logical AND operator
:59	TCMIOR	Stack logical inclusive OR operator
:5A	TCMXOR	Stack logical exclusive OR operator
:5B	TCMNOT	Stack logical NOT (monadic) operator
:5C	TCMEQ	Stack relational equate operator
:5D	TCMNE	Stack relational not equal operator
:5E	TCMGT	Stack relational greater than operator
:5F	TCMLT	Stack relational less than operator
:60	TCMGE	Stack relational greater than or equal operator
:61	TCMLE	Stack relational less than or equal operator
:70	TCMADD	Stack Add operator
:71	TCMSUB	Stack Subtract operator
:72	TCMMPY	Stack Multiply operator
:73	TCMDVD	Stack Divide operator

The stack operators operate on the linker reverse polish stack. Type codes :56 and :5B are monadic operators which cause the top element of the stack to be removed and the result of the operation to be deposited back on the stack. All other type codes are diadic operators. These type codes cause two elements to be taken off the stack, with the operation yielding a single result as for monadic operations. The stack operations are described under type code :62 below.

Type Code :62

Name TCKOFD

Layout K

Meaning: Field from stack

Type code :62 provides a termination for a sequence of stack operations begun with a begin stack :6A type code. When a :62 type code is encountered, the top element of the stack is removed, shifted as indicated by the shift count (+ is left, - is right, 0 is no shift), and ANDed with the MASK. The result is placed in the current position of the location counter. The location counter is incremented by one. Type codes :06, :07, :64, :74, and :75 have a different effect when placed between a begin stack :6A type code and a :62 type code. Their effect is to cause the value of the operand concerned to be pushed onto the linker reverse polish stack rather than to be deposited at the position of the location counter.

Type Code:63

Name TCINTA

Layout G

Meaning: Intermediate access, absolute

Type Code :64

Name TCINTR

Layout G

Meaning: Intermediate access, relocatable

Type code :64 is used to produce as data a reference to a location in another segment without referring to an external reference. The data stored is determined as R+VALUE, where R is the base of relocation of the segment associated with index INDEX in this program unit. If this type code is placed in a legal stack expression (i.e., a :66-:67 or :68 sequence or a :6A-:62 sequence), the data generated is placed on the linker reverse polish stack. In any other context this type code causes the data to be deposited at the current value of the location counter and the location counter to be incremented.

Type Code :66

Name TCZOP

Layout A

Meaning: Stack Definition

Type code :66 allows a symbol to be defined as a result of a linker reverse polish stack evaluation, instead of from the current value of the program counter. The format is:

Type code :66

:
:

Legal Stack Expression in Linker Reverse Polish Form

:
:

Type code :67 or :68

The trailing symbol, described by the :67 or :68 type code, is assigned the value which results from the stack evaluation.

Type Code :67

Name TCDENP

Layout B

Meaning: Primary Symbol Definition

Type code :67 causes the primary definition indexed by INDEX in the symbol list generated by type code :4A to be defined. If it is used as a termination of a linker reverse polish expression (as described under type code :66 above), the definition is derived from the top element of the stack. In any other use the defined value is taken to be that of the current value of the location counter.

Type Code :68

Name TCDENS

Layout B

Meaning: Secondary Symbol Definition

Type code :68 is like type code :67 above except that a secondary definition is defined and the INDEX field refers to a symbol list generated by a :4C type code.

Type Code :69

Name TCLOC

Layout B

Meaning: Location Counter Change

Type code :69 is used to change the location counter from referencing one segment to another. The INDEX is an index into the list generated by the :4F type code. This type code should be followed immediately by either origin type codes :04, :05 or by end program type codes :02, :03. If followed by :04, :05 type codes, the value of the location counter in the indexed segment is defined. If followed by :02, :03 type codes, the start address of the program is defined to be in the indexed segment.

Type Code :6A

Name TCKOP

Layout A

Meaning: Begin Stack operation

Type code :6A indicates the start of a stack expression which will be terminated by a :62 type code, and will cause data to be stored at the position of the location counter. For details see type code :62.

Type Code :6B

Name TCNCHP

Layout B

Meaning: Chain Element

Type code :6B provides an element in a chain linked list. At link or load time, each new Chain Element Record causes a word to be written which is the address of the previous Chain Element TC Record element. The first Chain Element Record is written as a zero. The INDEX refers to the Chain Element list as generated by the :4E type code.

Type Code :6C

Name TCHCHP

Layout B

Meaning: Chain Head

Type code :6C provides a head for a chain linked list. At link or load time, a word is written which is the address of the last element in the linked list. The INDEX refers to the Chain Element list as generated by the :4E type code.

Type Code :6D

Name TCNOTE

Layout J

Meaning: Note Record

Type code :6D specifies BYTES of commentary information and does not affect the operation of the Linker. TCNOTES may optionally be printed and/or copied into the output file(s).

Although no inherent meaning is associated with TCNOTE type codes, they are divided into three classes for processing by the Linker. A particular TCNOTE typecode is assigned to a class based on the byte immediately following the count. The classes are listed below.

<u>CLASS</u>	<u>DEFINITION</u>
FIRST BYTE 0	This class is used by language translators to insert their names and revision into the object text. The Linker will delete (unless the COPYXLATID LINK option is specified) and will not print this class (unless the XLATID LINK option is specified).
FIRST BYTE :80...:FF	This class has no meaning to the Linker. The Linker will copy (unless the NOCOPYOBJ NOTES LINK option is specified) and will not print this class (unless the OBJNOTES LINK option is specified). If this class is printed (OBJNOTES), the hexadecimal representation of the data will be printed.
FIRST BYTE :01...:7F	This class is assumed to be commentary and will be deleted (unless the COPYNOTES LINK option is specified) and copied and printed (unless the NONOTES LINK option is specified) by the Linker.

Type Code :6F

Name TCSGML

Layout H

Meaning: List Segment Attributes

Type code :6F is used in conjunction with the :4F type code to specify the length of, and the flags associated with, each segment of the list generated by type code :4F. If the number of segments generated by the :4F type codes is N, then BYTES must be 3*N. The data for each segment is a two-byte VALUE and a one-byte FLAGS. The VALUE is the length (in words) of that segment and the FLAGS are as defined below. Type code :6F is mandatory and must follow the :4F type code group, with which it is associated. The defined flags are:

<u>Bit</u>	<u>Meaning if set</u>
0	ROMable
1	Common
2	Sharable

The FLAGS byte can assume the following meanings:

Bits: 2 1 0

0 0 0	Non Sharable Data
0 0 1	Non Sharable Data in ROM
0 1 0	Common Data
0 1 1	Common Data in ROM
1 0 0	Sharable Data
1 0 1	Sharable Data in ROM
1 1 0	Global Common
1 1 1	Global Common in ROM

Type Code :74

Name TCTEXP

Layout B

Meaning: Text External (primary)

The INDEX in type code :74 refers to the symbol list generated by the :4B type code. A word is generated which is an address of a primary external symbol. If this type code is placed in a legal stack expression (i.e., a :66-:67 or :68 sequence, or a :6A-:62 sequence) then the word generated is placed on the stack. In any other context type code :74 causes a word to be deposited at the current value of the location counter and the location counter to be incremented.

Type Code :75

Name TCTEXS

Layout B

Meaning: Text External (secondary)

Type code :75 has the same effect as type code :74 except that the symbol referenced is a secondary external symbol and the INDEX refers to the list generated by type code :4D.

Type Code :77

Name TCNTSK

Layout L

Meaning: Task Execution Instances (Reserved)

Type Code :78

Name TCDADR

Layout M

Meaning: Task Y-Pool Generation (Reserved)

