

IB-19B005

DELTAMAX MOTION CONTROL SYSTEM

JANUARY 1998

DELTAMAX DEVICENET

INSTRUCTION BOOK

INDUSTRIAL INDEXING SYSTEMS, Inc.

Revision - A
Approved By:

Proprietary information of Industrial Indexing Systems, Inc. furnished for customer use only.
No other uses are authorized without the prior written permission of
Industrial Indexing Systems, Inc.

TABLE OF CONTENTS

| | |
|--|-----------|
| INTRODUCTION..... | 3 |
| DELTAMAX OBJECT MODEL..... | 4 |
| CONNECTING DELTAMAX TO DEVICENET NETWORK..... | 5 |
| DELTAMAX OBJECTS | 7 |
| FLAGS..... | 7 |
| Flag Object Definition | 8 |
| Instance Definitions | 8 |
| DATA MEMORY..... | 10 |
| Data Memory Object Definition | 10 |
| Data Memory Behavior..... | 11 |
| INTEGER | 12 |
| Integer Variable Memory Object Definition | 12 |
| Integer Variable Behavior..... | 12 |
| FLOAT | 13 |
| Float Variable Memory Object Definition..... | 13 |
| Float Variable Behavior | 13 |
| DELTAMAX PROGRAMMABLE PARAMETER OBJECT | 14 |
| I/O CONNECTIONS..... | 16 |
| Polled Connection | 16 |
| Bit Strobed Connection..... | 16 |
| MACROPROGRAM FEATURES..... | 17 |
| can_set..... | 17 |
| dnet_p_code..... | 18 |
| dnet_range | 19 |
| if_dnet_on..... | 19 |
| if_dnet_off | 19 |
| set_dnet | 20 |

| | |
|--|-----------|
| clr_dnet | 20 |
| dnet_flag_pm | 21 |
| dnet_int_pm | 22 |
| dnetflt_pm | 23 |
| Example MacroProgram(without Parameter Object) | 24 |
| Example MacroProgram(with Parameter Object)..... | 27 |
| EDS FILES..... | 29 |
| EDS File Example | 29 |
| APPENDIX A: DELTAMAX DEVICE DATA | 31 |

INTRODUCTION

This manual is intended to describe the DeviceNet Protocol interface implemented in the DeltaMax controller. The reader is expected to have some knowledge of the DeviceNet Protocol and nomenclatures.

DeviceNet is a low-cost communications link to connect industrial devices (such as limit switches, photoelectric switches, valve manifolds, motor starters, operator interfaces, ect..) as well as control devices to a network. The Open DeviceNet Vendor Association Publishes a DeviceNet Product Catalog three or four times a year for a current product listing of DeviceNet compatible devices.

DeviceNet is a trademark of the ODVA Open DeviceNet Vendor Association, inc.
8222 Wiles Road Phone: (305)340-5412
Suite 287 Fax:(305)340-5413
Coral Springs , FL 33067 <http://www.industry.net/ODVA>

IIS Industrial Indexing Systems Inc. is a member of ODVA. IIS maintains a Vendor ID with ODVA of 89.

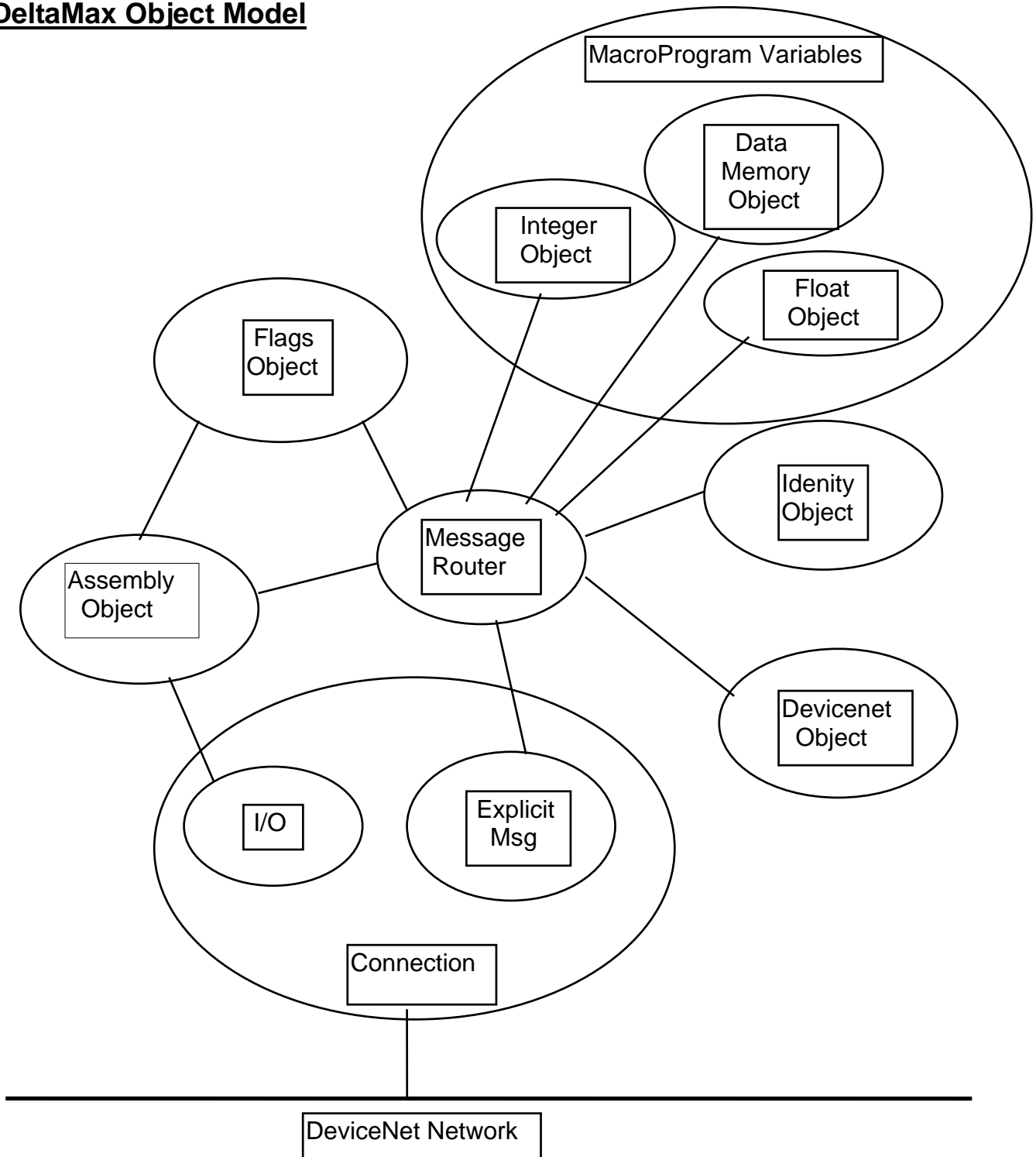
IIS has given the DeltaMax the following:

| | |
|--------------------|----------|
| Device Profile: | Generic |
| Product Name: | DeltaMax |
| Product Code: | 1 |
| Produuct Revision: | 1 |

Appendix A contains DeviceNet required object definitions for further DeltaMax product information.

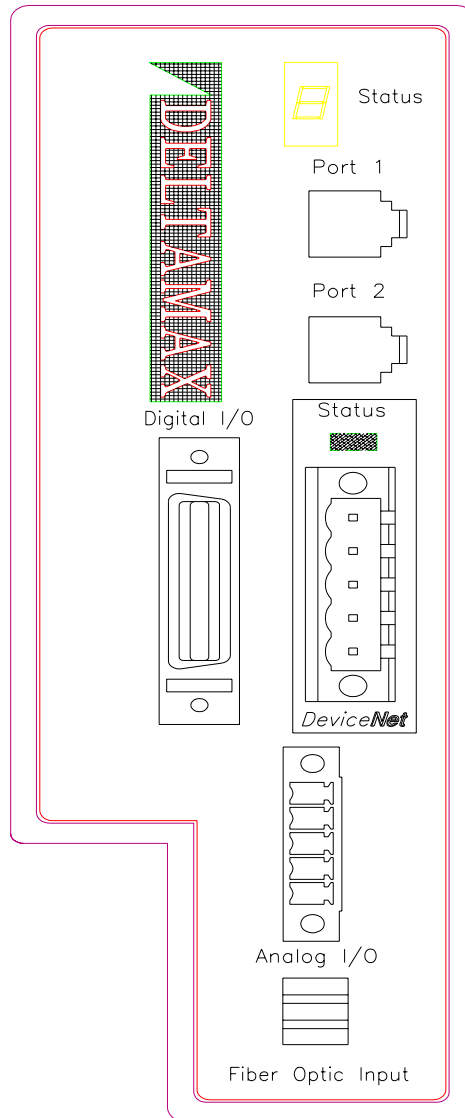
DeltaMax is a programmable position controller integrated with a motor amplifier. An application specific motion program is herein refered to as the MacroProgram. MacroPrograms are developed using IIS's "MacroPro II Toll Kit". DeviceNet offers the system designer and programmer the means to externally access and control the MacroProgram to achive their unique application requirements.

DeltaMax Object Model

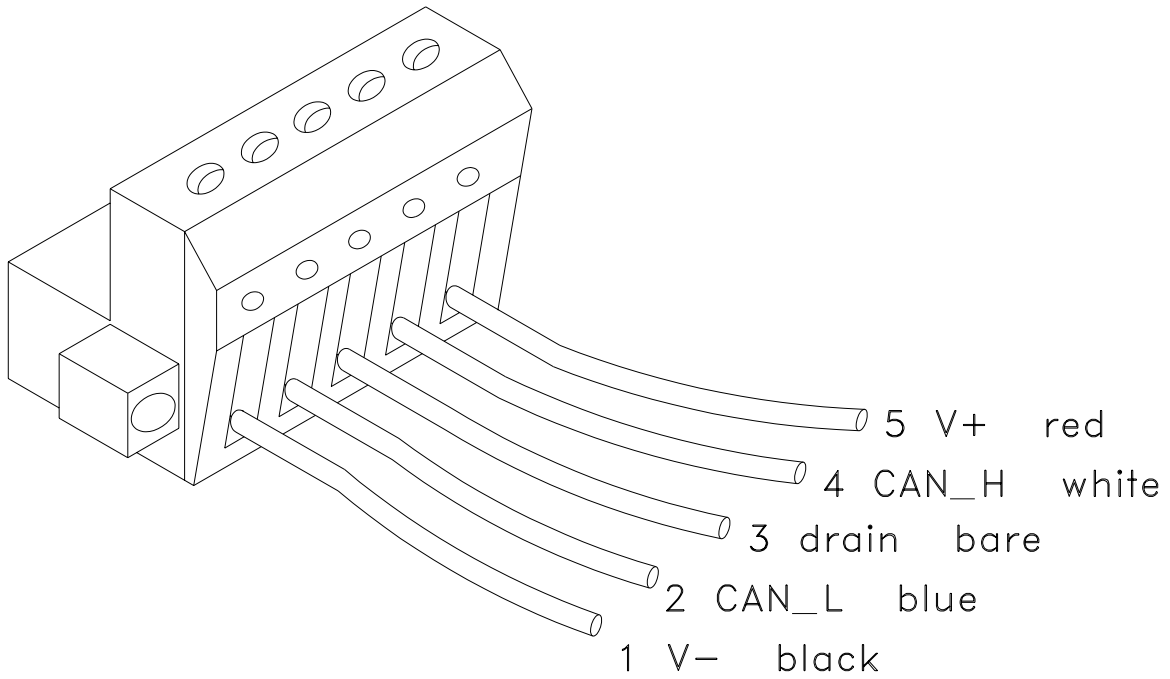


Connecting DeltaMax to DeviceNet Network

The figure below shows the location of the DeviceNet connector on the DeltaMax.



The figure below shows the open pluggable terminal connector required for the DeviceNet Network connections.



DELTAMAX OBJECTS

FLAGS

DeviceNet has limited access to DeltaMax FLAG bit map memory. FLAG memory in a DeltaMax contains vital information for the programmer developing a MacroProgram for any application. This bitmap is broken down as follows:

| flag # | description |
|---------|--|
| 0-15 | Discrete Inputs; Application specific, user programmable, originate from an external connector. |
| 16-23 | Discrete Outputs; Application specific, user programmable, output to an external connector. |
| 24-31 | Software programmable limit switches. |
| 32-55 | DeviceNet input flags. The DeviceNet port may read/write to these flags. To be considered as inputs only by a MacroProgrammer, therefore not write-able by a MacroProgram. |
| 56-71 | DeviceNet output flags. The DeviceNet port may only read these flags. To be considered as outputs from the MacroProgram, they may be written too or read from in a MacroProgram. |
| 72-79 | Timers |
| 80-95 | Axis1 status flags; example JOGGING,BUSY,DOWN,INDEXING, ect.. |
| 96-111 | Pseudo Axis status flags, similar to Axis1 status flags, however this is a software axis |
| 112-127 | Analog status flags. |
| 128-132 | DeviceNet Status flags; status flags read only. The DeviceNet port may only read these flags. |
| 133-207 | Not Used |
| 208-255 | User flags; have programmable effects in motion program. |

Flag Object Definition

Class Code: 100 (64HEX)

Instances: 1-256, one instance for each flag. Flags are zero based therefore instance 1 is for flag zero.

Instance Attributes: (one only)

| Attr. ID | Access Rule | Attr. Name | Data Type | Description |
|----------|-------------|------------|-----------|-----------------------------|
| 3 | get/set | Value | BOOL | The logic value of the flag |

The access rule for each instance is dependant on the flag# associated with it as follows:

| Instance | Flag# | Access Rule |
|----------|---------|-------------|
| 1-32 | 0-31 | none |
| 33-56 | 32-55 | get/set |
| 57-72 | 56-71 | get |
| 73-128 | 72-127 | none |
| 129-144 | 128-143 | get |
| 145-256 | 144-255 | none |

Instance Definitions

33 thru 56) These instances are DeviceNet input flags, therefore they are read-only MacroProgram flags. A device external to the DeltaMax may set and clear these flags over DeviceNet.

57 thru 72) These instances are DeviceNet output flags, therefore they may be set or cleared in a MacroProgram. A device external to DeltaMax may access their attributes over DeviceNet.

129) CanBus Enabled Status Flag. This instance is TRUE once a canset instruction is succesfully executed in a MacroProgram.

130) CanBus Fault Flag. This instance is TRUE if DeviceNet interface goes busOff.

131) Explicit Connection Exists. This instance is set while an explicit connection is present.

- 132) Polled Connection Exists. This instance is set while a polled connection is present.
- 133) Bit-Strobed Connection Exists. This instance is set while a bit-strobed connection is present.
- 134) DeviceNet Bus Power Down. This instance is set when the DeltaMax hardware detects that the DeviceNet Network power has faulted. The flag is intended for MacroProgram use.

DATA MEMORY

This object in the DeltaMax controller allows access to all of the MacroProgram variable memory area. It offers addressing modes that the individual Float and Integer Objects do not. It is intended to facilitate large block transfers of memory.

MacroProgram variable memory is set aside exclusively for an individual application. In total it is 16000 bytes, however it is broken into two fairly unique sections. One section holds data that is referred to in a MacroProgram of type integer, they are 32bits wide. The other section holds type floats, they are 64bits wide. The two sections have beginning address offsets of 0 and an ending offset depending on the application. The integer and float sections of data memory are contiguous, therefore the float address offset of 0 is actually address following the integer ending offset. Variable address offsets are easily obtained by the programmer in a symbol table file generated by the "MacroPro II" tool kit.

Data Memory Object Definition

Class Code: 103 (67HEX)

Instances: 1- 2 (1-Integers, 2-Floats)

Instance Attributes:

| Attr. ID | Access Rule | Attr. Name | Data Type | Description |
|----------|-------------|----------------|--------------|---|
| 1 | get/set | Address Offset | UINT | Address offset used for Attr. 3 when accessing the value. Default = 0. |
| 2 | get/set | Data Length | UCHAR | Data access length 1..8 Defaults: 4 for integers instance 1 8 for floats instance 2 |
| 3 | get/set | Value | DINT LINT | The data memory value. The data type can be modified with attribute 2. |
| 4 | get/set | Address Mode | UCHAR | Available for selecting an addressing mode, see table 1.1 Default = 0, address is in service data. |
| 5 | get | Unprotected Lo | UINT | Data is accessible for write above this offset. Default = 0 |
| 6 | get | Unprotected Hi | UINT | Data is accessible for write below this offset. Default = ending offset, application specific. |

Table 1 - Addressing Modes (attribute 4)

| Value | Description |
|-------|---|
| 0 | Address offset is obtained from service data. |
| 1 | Address is obtained from the address offset attribute, the attribute is not incremented after the access. |
| 2 | Address is obtained from the address offset attribute, with auto-increment, in that the address offset is incremented after the access by the "Data Length". Once the address reaches the end offset a Vendor Specific error is returned with add code of 01 (address offset out of range). |

Data Memory Behavior

Attributes 5 and 6 are defaulted such that over DeviceNet MacroProgram variables may not be written to. If it is required that variables in a MacroProgram be written to over the Network then a **dnet_range** instruction must be executed in the MacroProgram, see **dnet_range** section herein.

INTEGER

The Integer Object allows easy access to Integer variables for most DeviceNet devices.

This object in the DeltaMax controller allows access to only the integers of the MacroProgram variable memory area. There is one instance for each integer of the MacroProgram. Due to the nature of DeviceNet, one is able to only address up to the first 256 instances of the Integer Memory space, however if need be one can access all of integer space using the Data Memory Object.

Integer Variable Memory Object Definition

Class Code: 101 (65HEX)

Instances: 1 thru 256

Instance Attributes:

| Attr. ID | Access Rule | Attr. Name | Data Type | Description |
|----------|-------------|----------------|-----------|-------------------------------------|
| 1 | get | Address Offset | UINT | Address offset of the instance. |
| 2 | get | Data Length | UCHAR | Always 4 bytes. |
| 3 | get/set | Value | DINT | Value of Integer variable instance. |
| 4 | get | Unprotected Lo | UINT | Same as Data Memory Object. |
| 5 | get | Unprotected Hi | UINT | Same as Data Memory Object. |

Integer Variable Behavior

Attributes 4 and 5 are defaulted such that over DeviceNet MacroProgram variables may not be written to. If it is required that variables in a MacroProgram be written to over the Network then a **dnet_range** instruction must be executed in the MacroProgram, see **dnet_range** section herein.

FLOAT

The Integer Object allows easy access to Integer variables for most DeviceNet devices.

This object in the DeltaMax controller allows access to only the floats of the MacroProgram variable memory area. There is one instance for each float of the MacroProgram. Due to the nature of DeviceNet, one is able to only address up to the first 256 instances of Float Memory space, however if need be one can access all of the float space using the Data Memory Object.

Float Variable Memory Object Definition

Class Code: 102 (66HEX)

Instances: 1 thru 256

Instance Attributes:

| Attr. ID | Access Rule | Attr. Name | Data Type | Description |
|----------|-------------|----------------|-----------|-----------------------------------|
| 1 | get | Address Offset | UINT | Address offset of the instance. |
| 2 | get | Data Length | UCHAR | Always 8 bytes. |
| 3 | get/set | Value | LREAL | Value of Float variable instance. |
| 4 | get | Unprotected Lo | UINT | Same as Data Memory Object. |
| 5 | get | Unprotected Hi | UINT | Same as Data Memory Object. |

Float Variable Behavior

Float variables are 64bits wide, they are defined in IEEE 754 for the basic double floating-point format.

Attributes 4 and 5 are defaulted such that over DeviceNet MacroProgram variables may not be written to. If it is required that variables in a MacroProgram be written to over the Network then a **dnet_range** instruction must be executed in the MacroProgram, see **dnet_range** section herein.

DeltaMax Programmable Parameter Object

The DeltaMax Parameter object is programmable via macro instructions. This allows for unique application specific requirements to be accessed over the DeviceNet network as a parameter. Tools like DeviceNet Manager by Allen-Bradley may then easily upload the parameter instances displaying their value and text description for user friendly interface to the controllers MacroProgram. DeviceNet Manager also has the capability to then generate a DeviceNet EDS File (electronic data sheet) from the information obtained from a Parameter Object Instance.

The Parameter Object Instance attributes are defined by ODVA as follows:

| Attribute Name (ID) | [Expected Values, Responses] Full Parameter Instance |
|----------------------------|--|
| Undefined (00) | [Attribute_Not_Supported (x14)] |
| Parameter_Value (01) | Depends on Attributes 5, 6 |
| Link_Path_Size (02) | [USINT (0..255)] |
| Link_Path (03) | [PATH_ID[Link_Path_Size] see Note 1 below] |
| Descriptor (04) | [WORD (0000000000xxxxxx)] |
| Data_Type (05) | [USINT (1..26)] |
| Data_Size (06) | [USINT (1..255)] |
| Parameter_Name_String (07) | [SHORT_STRING with max of 16 characters] |
| Units_String (08) | [SHORT_STRING with max of 4 characters] |
| Help_String (09) | [SHORT_STRING with max of 64 characters] |
| Minimum_Value (10) | [Same type as Parameter_Value (01)] |
| Maximum_Value (11) | [Same type as Parameter_Value (01)] |
| Default_Value (12) | [Same type as Parameter_Value (01)] |
| Scaling_Multiplier (13) | [UINT (1..65535)] |
| Scaling_Divisor (14) | [UINT (1..65535)] |
| Scaling_Base (15) | [UINT (0..65535)] |
| Scaling_Offset (16) | [INT (-32768..32767)] |
| Multiplier_Link (17) | [UINT (1..65535)] |
| Divisor_Link (18) | [UINT (1..65535)] |
| Base_Link (19) | [UINT (0..65535)] |
| Offset_Link (20) | [UINT (0..65535)] |
| Decimal_Precision (21) | [USINT (0..255)] |
| Undefined (22..99) | [Attribute_Not_Supported] |

Parameters may be programmed for a Flag, Integer, or Float Object instances. Instances of the Flag, Integer and Float Objects are just variables declared in a MacroProgram. Once the variables are declared in a motion application program then the following MacroProgram instructions may be added to the program to generate a parameter object for DeviceNet Manager Tools:

dnet_flag_pm
dnet_int_pm
dnetflt_pm

The instructions to program a parameter object must be contained within a **begin_cfg** and **end_cfg** instruction block, also a **begin_cfg** instruction in a MacroProgram must be the first executable instruction in a program. The usage of the above instructions is defined in the MacroProgram Features section of this document, also refer to the example programs.

The **dnet_range** instruction must be executed prior to a **dnet_int_pm** or **dnetflt_pm** for any parameters to be any thing but **READ ONLY**. This also effects how parameters behave within any DeviceNet managing tools.

It is recommended that a parameter object be programmed into a DeltaMax prior to enabling the DeviceNet port. the DeviceNet port is enabled once a **can_set** instruction is executed.

I/O Connections

The DeltaMax has two possible static I/O instances of the Connection Object, see appendix A, one Polled (Instance 2) and one Bit-Strobed (Instance 3).

Polled Connection

The DeltaMax will consume 3 bytes in a polled command. These three bytes will set or clear flag numbers 32 thru 55, instances 33 thru 56 resp., of the Flags Object as follows.

| | Bit Numbers | | | Bit Numbers | | | Bit Numbers | | |
|-------|-------------|--------|----|-------------|--------|----|-------------|--------|----|
| | 7 | BYTE 1 | 0 | 7 | BYTE 2 | 0 | 7 | BYTE 3 | 0 |
| flag# | 39 | | 32 | 47 | | 40 | 55 | | 48 |

The DeltaMax will produce 2 bytes in the polled response, the bytes will contain the present condition of flag numbers 56 thru 71, instances 57 thru 72, of the Flags Object as follows.

| | Bit Numbers | | | Bit Numbers | | |
|-------|-------------|--------|----|-------------|--------|----|
| | 7 | BYTE 1 | 0 | 7 | BYTE 2 | 0 |
| flag# | 63 | | 56 | 71 | | 64 |

Bit Strobed Connection

The DeltaMax shall produce only from a Bit-Strobed command. The response shall contain the present state of flag numbers 56 thru 71. The Data bytes in the response are identical to the Polled connection response as follows.

| | Bit Numbers | | | Bit Numbers | | |
|-------|-------------|--------|----|-------------|--------|----|
| | 7 | BYTE 1 | 0 | 7 | BYTE 2 | 0 |
| flag# | 63 | | 56 | 71 | | 64 |

Important Note: A Bit-Strobed connection when allocated requires the DeltaMax processor to listen to all Group 2 DeveNet traffic on the bus. Under normal bus conditions this should not be a significant borden, however it could possibly slow MacroProgram processing under very heavy DeviceNet traffic.

MACROPROGRAM FEATURES

can_set

The DeviceNet MacID and Baudrate are configured in the MacroProgram via the “can_set” instruction.

can_set arg1,arg2,arg3

arg1: DeviceNet ID
MacroProgram Value Type: integer variable, integer constant.
Value Ranges: 0-63, and -1

arg2: DeviceNet Baudrate in kbaud.
MacroProgram Value Type: integer variable, integer constant.
Value Ranges: 125,250,500, and -1

arg3: Protocol setting.
MacroProgram Value Type: integer variable, integer constant.
Value Ranges: 2
Note: values 0,1 are reserved for IIS CanBus protocol.

If arg1 and/or arg2 are set to -1 then that DeviceNet parameter may be set over the DeviceNet network. The DeltaMax may then be configured by a software tool such as DeviceNet Manager by Allen-Bradley. Once the parameter is configured it is stored in Nonvolatile memory so that it is remembered by the controller from one powerup to the next.

A **can_set** executed after a CanBus fault flag (flag# 129) will attempt to clear the fault and bus-off condition so that communications may continue.

A **can_set** instruction may reside within a begin_cfg and end_cfg instruction block so that once a MacroProgram is loaded, the DeltaMax executes its DeviceNet Network Access State Machine. This allows for a DeltaMax to be online without having the MacroProgramming running.

IMPORTANT: The DeltaMax doesn't attempt to access a DeviceNet Network until a **can_set** is executed with correct arguments.

dnet_p_code

SYNTAX:

label **dnet_p_code** product_code,text_label

PARAMETERS:

| <u>Name</u> | <u>Description</u> | <u>Range</u> | <u>Type Allowed</u> |
|--------------|--|--------------|---------------------|
| product_code | DeviceNet product code number in the DeltaMax Identity Object. | 1 to 65535 | IC |
| text_label | Label of a text string to describe the deltamax application program. | | T |

DESCRIPTION:

This instruction must reside in a program between a **begin_cfg** and **end_cfg** instruction at the top of a MacroProgram.

Assigns a "DeviceNet Product Code" in the Identity Object of the controller. It also will change the "DeviceNet Product Type" in the Identity Object of the controller from 0Hex to 300Hex, so that the Product Type changes from "Generic" to "vendor_specific".

The instruction is intended for giving different controllers on the same DeviceNet network unique identities. This is especially helpful when the controllers have different programs and application parameters to accomplish different task in a system having one DeviceNet Network.

RETURNS:

None.

dnet_range

Allows the maroprogrammer to set limits on what variables in the MacroProgram may be written to by the DeviceNet port.

dnet_range argument_1,argument_2

arg1: lower boundry.

Values: any variable label within the MacroProgram.

arg2: upper boundry.

Values: variable label in a sequecuallly higher order in memory than arg1.

IMPORTANT: The **dnet_range** instruction should be executed prior to the parameter object programming instructions **dnetflt_pm** and **dnetint_pm**, so that the parameter may be written to over the DeviceNet port if desired. The **dnet_range** instruction may therefore reside within a **begin_cfg** and **end_cfg** instruction.

if_dnet_on

Allows condition checks on the DeviceNet flags 32-71, if the flag is on the branch is executed.

if_dnet_on flag#,address_label

flag#: devicenet flag number to be tested.

MacroProgram Value Type: integer equate, integer constant.

Value Ranges: 32-71

address_label: instruction label within MacroProgram.

if_dnet_off

Allows condition checks on the DeviceNet flags 32-71, if the flag is off the branch is executed.

if_dnet_off flag#,address_label

flag#: devicenet flag number to be tested.

MacroProgram Value Type: integer equate, integer constant.

Value Ranges: 32-71

address_label: instruction label within MacroProgram.

set_dnet

Allows setting of a Devicenet flag from 56-71, sets the value of the flag boolean level true or one.

set_dnet **flag#**

flag#: devicenet flag number to be set.
MacroProgram Value Type: integer equate, integer constant.
Value Ranges: 56-71

clr_dnet

Allows setting of a Devicenet flag from 56-71, sets the value of the flag boolean level false or zero.

set_dnet **flag#**

flag#: devicenet flag number to be cleared.
MacroProgram Value Type: integer equate, integer constant.
Value Ranges: 56-71

dnet_flag_pm

SYNTAX:

label **dnet_flag_pm** dnet_flag,text_label

PARAMETERS:

| <u>Name</u> | <u>Description</u> | <u>Range</u> | <u>Type Allowed</u> |
|-------------|---|--------------|---------------------|
| dnet_flag | DeviceNet I/O flag number | 32-71 | IC |
| text_label | Label of a text string, string length is 16 characters maximum. | | T |

DESCRIPTION:

This instruction must reside in a program between a **begin_cfg** and **end_cfg** instruction at the top of a MacroProgram.

Assigns a parameter to the DeviceNet programmable Parameter Object of the controller. The parameter assigned is the instance of the Flag Object associated with the dnet_flag argument of the instruction. During instruction execution the MacroProgram variable shall be given the default value of the instruction.

Instances of the Parameter Object are arranged in the order of which they are assigned in a program. No more than 16 assignments may be made in a program, in that their can only be 16 instances of the Parameter Object. That includes assignments from instructions **dnet_intg_pm** and **dnetflt_pm**.

RETURNS:

None.

dnet_int_pm

SYNTAX:

label **dnet_intg_pm** value,text_label,min,max,default

PARAMETERS:

| <u>Name</u> | <u>Description</u> | <u>Range</u> | <u>Type Allowed</u> |
|-------------|---|---------------------------|---------------------|
| value | The declared integer value to use for a DeviceNet parameter. | | IV |
| text_label | Label of a text string to describe the parameter, string length is 16 characters maximum. | | T |
| min | Minimum intended value for the assigned parameter. | 2147483647 to -2147483647 | IC |
| max | Maximum intended value for the assigned parameter. | 2147483647 to -2147483647 | IC |
| default | Default value for the assigned parameter. | 2147483647 to -2147483647 | IC |

DESCRIPTION:

This instruction must reside in a program between a **begin_cfg** and **end_cfg** instruction at the top of a MacroProgram.

Assigns a parameter to the DeviceNet programmable Parameter Object of the controller, see IB-19B005. The parameter assigned is the instance of the Integer Object associated with the integer value argument of the instruction. During instruction execution the MacroProgram variable shall be given the default value of the instruction.

Instances of the Parameter Object are arranged in the order of which they are assigned in a program. No more than 16 assignments may be made in a program, in that their can only be 16 instances of the Parameter Object. That includes assignments from instructions **dnet_flag_pm** and **dnetflt_pm**.

IMPORTANT: The **dnet_range** instruction should be executed prior to this instruction, so that the parameter may be written to over the DeviceNet port if desired. Otherwise the parameter shall have a READ ONLY attribute.

RETURNS:

None.

dnet_flt_pm

SYNTAX:

label **dnet_flt_pm** value,text_label,min,max,default

PARAMETERS:

| <u>Name</u> | <u>Description</u> | <u>Range</u> | <u>Type Allowed</u> |
|-------------|--|------------------------------|---------------------|
| value | The declared integer value to use for a DeviceNet parameter. | | IF |
| text_label | Label of a text string to describe the parameter, string length is 16 characters maximum.. | | T |
| min | Minimum intended value for the assigned parameter. | IEE754 double floating-point | FC |
| max | Maximum intended value for the assigned parameter. | IEE754 double floating-point | FC |
| default | Default value for the assigned parameter. | IEE754 double floating-point | FC |

DESCRIPTION:

This instruction must reside in a program between a **begin_cfg** and **end_cfg** instruction at the top of a MacroProgram.

Assigns a parameter to the DeviceNet programmable Parameter Object of the controller. The parameter assigned is the instance of the Float Object associated with the float value argument of the instruction. During instruction execution the MacroProgram variable shall be given the default value of the instruction.

Instances of the Parameter Object are arranged in the order of which they are assigned in a program. No more than 16 assignments may be made in a program, in that their can only be 16 instances of the Parameter Object. That includes assignments from instructions **dnet_flag_pm** and **dnet_intg_pm**.

IMPORTANT: The **dnet_range** instruction should be executed prior to this instruction, so that the parameter may be written to over the DeviceNet port if desired. Otherwise the parameter shall have a READ ONLY attribute.

RETURNS:

None.

Example MacroProgram(without Parameter Object)

```

=====
! MacroPro II for Windows program - DeltaMax with DeviceNet Example
!
! TITLE: DNET.PRG
!
! DESCRIPTION: Example program for DeviceNet users
!
! EQUIPMENT: DeltaMax, MACPRO_II
!
! REVISION HISTORY:
!
! REV  WHO   DATE   DESCRIPTION
! ----  ----  -
!  0    ABC   01/01/97 Initial release
!
=====
                                declare      on

DNET          equ          2
MOTOR         equ          1
BUSY          equ         78
DOWN         equ          77
OFF           equ          67
MOTOR_OFF     equ         (MOTOR*16)+OFF
MOTOR_BUSY    equ         (MOTOR*16)+BUSY
MOTOR_DOWN    equ         (MOTOR*16)+DOWN

DNET_ENABLE   equ          128      lequate for the DeviceNet enabled flag
DNET_FAULT    equ          129      lequate for the DeviceNet fault flag
DNET_INPUT1   equ           32      lequate for DeviceNet input 1 for do trackspd
DNET_OUTPUT1  equ           56      lequate for DeviceNet output 1 for MOTOR_BUSY

!***** variables to have write access over DeviceNet
spd           integer
acdc          integer
multplr       integer
divisor       float

!**** delta_comp variables
lim           integer
pg            integer
ffwd         integer

!**** motor axis variables
com           integer
pos          integer

!**** can_set variables
canid        integer
canbaud      integer
canarb       integer

!**** application variables
angle        integer
div_angle    float
mult_angle   integer
can_faults   integer

!*****PROGRAM VARIABLE INITIALIZATIONS*****
      let          spd=100
      let          acdc=200
  
```

```

let divisor=1
let multplr=1

let lim=1
let pg=50
let fwd=50

let canid=-1
let canbaud=-1
let canarb=DNET
    
```

!*****DEVICENET INITIALIZATION*****

```

dnet_range spd,divisor
can_set canid,canbaud,canarb
wait_dneten if_stat_off 128,wait_dneten ! verify devicenet enabled
    
```

!*****MOTOR AXIS INITIALIZATION*****

```

drive_on MOTOR
set_speed MOTOR,spd
set_ac_dc MOTOR,acdc
    
```

!*****SOFTWARE INTERRUPT INITIALIZATIONS*****

```

swi_if_on 0,DNET_FAULT,CanFaulted
enable_swi
    
```

!*****MAIN PROGRAM Loop*****

```

Main if_dnet_on DNET_INPUT1,run_trackspeed

Gosub ForceDecel
Goto calcs

run_trackspeed gosub DoTrackSpeed

calcs get_com MOTOR,com
get_pos MOTOR,pos

let angle=pos&4095 ! get present angle from pos
let div_angle=angle/divisor ! divide angle
let mult_angle=angle*multplr ! multiple angle

if_stat_on MOTOR_BUSY,set_out1
clr_dnet DNET_OUTPUT1
goto Main
set_out1 set_dnet DNET_OUTPUT1
goto Main
    
```

!*****SUBROUTINES*****

!*****

!*** Force Decel Subr.*****

```

ForceDecel if_stat_off MOTOR_BUSY,skipfdecel
f_decal MOTOR
skipfdecal return_sub
    
```

!*** Track Speed Subr.*****

```

DoTrackSpeed if_stat_off MOTOR_OFF,gotrack
drive_on MOTOR

gotrack set_ac_dc 1,acdc
track_spd MOTOR,spd
    
```

return_sub

```
!*****  
!*****INTERRUPT ROUTINES*****  
!*****
```

!*** CanBus faulted

!*** Description: Tries to clear bus_off condition due to a
! possible DeviceNet Network error, for a
! limited number of times.

```
CanFaulted      disable_swi  
                 let          can_faults=can_faults+1  
                 if          can_faults>10,skipcanset  
                 can_set     canid,canbaud,canarb  
skipcanset      enable_swi  
                 return_sub
```

Example MacroProgram(with Parameter Object)

```

=====
!
! MacroPro II for Windows program - DeviceNet Parameter Object Example
!
! TITLE: DNET.PRG
!
! DESCRIPTION: Example program for DeviceNet users
!
! EQUIPMENT: DeltaMax, MACPRO_II
!
! REVISION HISTORY:
!
! REV  WHO  DATE  DESCRIPTION
! ----  ----  ----  -
!  0    ABC  01/01/98 Initial release
!
=====

                declare      on

DNET                equ      2

MOTOR               equ      1
BUSY                equ      78
DOWN                equ      77
OFF                 equ      67
DRV1_OFF            equ      83
MOTOR_OFF           equ      (MOTOR*16)+OFF
MOTOR_BUSY          equ      (MOTOR*16)+BUSY
MOTOR_DOWN          equ      (MOTOR*16)+DOWN

DNET_ENABLE         equ      128      !equate for the DeviceNet enabled flag
DNET_FAULT          equ      129      !equate for the DeviceNet fault flag
DNET_INPUT1         equ      32       !equate for DeviceNet input 1 for do trackspd
DNET_INPUT2         equ      33       !equate for DNet Input 2 to turn drive off
DNET_OUTPUT1        equ      56       !equate for DeviceNet output 1 for MOTOR_BUSY

RUN_BIT             equ      255

! variables to have write access over DeviceNet
spd                 integer
acdc                integer
time               integer
multplr            float
divisor            float

! motor axis variables
com                 integer
pos                 integer
time_strt          integer
get_tm             integer
tmp_tm             integer

run_text           text      "run input"
div_text           text      "float divisor"
spd_text           text      "speed in rpm"
acdc_text          text      "accel"
time_text          text      "runtime (msec)"
ID_text            text      "Dnet Eamle"
DrvOff_text        text      "MonetaryDriveOff"
  
```

!*****DEVICENET PARAMETER INITIALIZATION*****

```

begin_cfg
dnet_p_code    10,ID_text    !this is the first executable instr.
dnet_range     spd,divisor   !define devicenet product code
dnet_int_pm    time,time_text,10,100,10 !define read/write memory for devicenet
dnet_ft_pm     divisor,div_text,1.0,100.0,10.5
dnet_int_pm    spd,spd_text,10,1000,100
dnet_int_pm    acdc,acdc_text,10,100,50
dnet_flag_pm   DNET_INPUT1,run_text
dnet_flag_pm   DNET_INPUT2,DrvOff_text
can_set        -1,125,2
end_cfg
  
```

!*****DEFINE SWI(s)*****

```

swi_if_on      0,DNET_INPUT1,TurnDriveOn
swi_if_off     2,DNET_INPUT1,Stop
swi_if_on      1,DNET_INPUT2,TurnDriveOff

enable_swi
  
```

!*****PROGRAM VARIABLE INITIALIZATIONS*****

```

let            multplr=1
get_time       time_strt
  
```

!*****PROGRAM START*****

```

drive_on      1

wait_dneten   if_stat_off    128,wait_dneten    ! verify devicenet enabled

loop          get_time       get_tm
              let           tmp_tm = get_tm - time_strt
              let           tmp_tm = tmp_tm * 5
              if_flag_on    RUN_BIT,Trackspeed
              let           time = 0
              goto          loop

Trackspeed    let           time = tmp_tm
              set_ac_dc     1,acdc
              track_spd     1,spd
              goto          loop

done          goto          done
  
```

!*****SWI: TurnDriveOn*****

```

TurnDriveOn   if_stat_off    DRV1_OFF,skip_drvon
              drive_on      1
skip_drvon    get_time       time_strt
              set_flag      RUN_BIT
              enable_swi
              return_sub

Stop          f_decal        1
              clr_flag      RUN_BIT
              enable_swi
              return_sub

TurnDriveOff  drive_off      1
              clr_flag      RUN_BIT
              enable_swi
              return_sub
  
```

EDS Files

The Electronic Data Sheet (EDS) allows a configuration tool to automate the DeviceNet device configuration process. An EDS contains vendor specific and/or MacroProgram specific information defined for a particular application. The EDS file may be used to easily get a DeltaMax up on the DeviceNet Network.

An EDS definition is available by contacting ODVA, or obtaining Section 2 Chapter 4 of the ODVA DeviceNet Specification.

The following EDS file example was generated using DeviceNet Manager and a simple text editor. DeviceNet Manager is a DeviceNet software tool distributed by Allen Bradley. The example is intended to work with the MacroProgram exemplified without a parameter object, herein.

If a parameter object is MacroProgrammed into the DeltaMax then a EDS file maybe generated automatically from a DeviceNet such as DeviceNet Manager. The tool extracts the necessary information from the DeltaMax, then writes the file to the PC.

EDS File Example

\$ DeviceNet Manager Generated Electronic Data Sheet
\$

[File]

[Device]

| | | |
|--------------------|--|------------------------|
| VendCode | = 89; | \$ Vendor Code |
| ProdType | = 0; | \$ Product Type |
| ProdCode | = 1; | \$ Product Code |
| | | |
| MajRev | = 1; | \$ Major Rev |
| MinRev | = 0; | \$ Minor Rev |
| VendName | = "Industrial Indexing Systems, Inc"; | |
| ProdTypeStr | = "Generic"; | |
| ProdName | = "DeltaMax"; | |
| Catalog | = "0"; | |

[IO_Info]

Default = 0X0000;

[ParamClass]

MaxInst=4;
Descriptor=0;
CfgAssembly=0;

[Params]

\$ Flags Object Parameters

| | | |
|--------------------------------|------------------------------------|---|
| Param1= | | \$ class:0x64,Inst:0x21,Attr:0x3 |
| 0, | \$ parameter value slot | |
| 6, "20 64 24 21 30 03", | \$ link path size and path | |
| 0x0002, | \$ descriptor: enumerated | |
| 4, | \$ data type: BOOLEAN | |
| 1, | \$ data size: 1 for BOOLEAN | |

```
"RUN INPUT",      $ parameter name
"",               $ units string
"",               $ help string
0, 1, 0,          $ min, max, default
1, 1, 1, 0,      $ mult, div, base, offset scaling
0, 0, 0, 0, 0 ;  $ scaling links and precision
```

\$ Integer Object Dynamic Parameters

```
Param2=           $ class:0x65,Inst:0x01,Attr:0x3
0,                $ parameter value slot
6, "20 65 24 01 30 03", $ link path size and path
0x0000,           $ descriptor: none
6,                $ data type: DINT
4,                $ data size: 4 for DINT
"Speed",          $ parameter name
"rpm",            $ units string
"",               $ help string
0, 3000, 100,     $ min, max, default
1, 1, 1, 0,      $ mult, div, base, offset scaling
0, 0, 0, 0, 0 ;  $ scaling links and precision
```

```
Param3=           $ class:0x65,Inst:0x03,Attr:0x3
0,                $ parameter value slot
6, "20 65 24 03 30 03", $ link path size and path
0x0000,           $ descriptor: none
6,                $ data type: DINT
4,                $ data size: 4 for DINT
"Multiplier",    $ parameter name
"",               $ units string
"",               $ help string
0, 4, 1,          $ min, max, default
1, 1, 1, 0,      $ mult, div, base, offset scaling
0, 0, 0, 0, 0 ;  $ scaling links and precision
```

```
Param4=           $ class:0x65,Inst:0x0d,Attr:0x3
0,                $ parameter value slot
6, "20 65 24 0d 30 03", $ link path size and path
0x0010,           $ descriptor: readonly
6,                $ data type: DINT
4,                $ data size: 4 for DINT
"Angle * Multiplier", $ parameter name
"bits(4096bits/rev)", $ units string
"",               $ help string
0, 4095, 0,       $ min, max, default
1, 1, 1, 0,      $ mult, div, base, offset scaling
0, 0, 0, 0, 0 ;  $ scaling links and precision
```

[EnumPar]

\$ Flags Object Parameters Enumerations

```
Param1=           $ Bit Definitions for Param 1
"OFF",
"ON";
```

[Groups]

APPENDIX A: DeltaMax Device Data

| Device Data | | | | | | | | |
|--|---|--|-------------------------------------|-------------------------------------|-------------------------------------|----------|--------------------------|----------|
| General Device Data | Conforms to Devicenet Specification | Volume I – Release | <u>1.3</u> | | | | | |
| | | Volume II – Release | <u>1.2</u> | | | | | |
| | Vendor Name | <u>Industrial Indexing Systems, Inc.</u> | | | | | | |
| | Device Profile Name | <u>Generic Device</u> | | | | | | |
| | Product Name | <u>DeltaMax</u> | | | | | | |
| | Product Catalog Number | <u>0</u> | | | | | | |
| Product Revision | <u>1</u> | | | | | | | |
| DeviceNet Physical Conformance Data | Network Power Consumption (Max) | <u>0.12</u> A@11V dc (worst case) | | | | | | |
| | Connector Style | Open–Hardwired | <input type="checkbox"/> | Sealed-Mini | <input type="checkbox"/> | | | |
| | | Open–Pluggable | <input checked="" type="checkbox"/> | Sealed-Micro | <input type="checkbox"/> | | | |
| | Isolated Physical Layer | Yes | <input checked="" type="checkbox"/> | | | | | |
| | | No | <input type="checkbox"/> | | | | | |
| | LEDs Supported | Module | <input type="checkbox"/> | Combo Mod/Net | <input checked="" type="checkbox"/> | | | |
| | | Network | <input type="checkbox"/> | I/O | <input type="checkbox"/> | | | |
| | MAC ID Setting | DIP Switch | <input type="checkbox"/> | Software Settable | <input checked="" type="checkbox"/> | | | |
| | | Other | <u>MacroProgrammable</u> | | | | | |
| | Default MAC ID | <u>63</u> | | | | | | |
| | Communication Rate Setting | DIP Switch | <input type="checkbox"/> | Software Settable | <input checked="" type="checkbox"/> | | | |
| | | Other | <u>MacroProgrammable</u> | | | | | |
| Communication Rates Supported | 125k bit/s | <input checked="" type="checkbox"/> | 500k bit/s | <input checked="" type="checkbox"/> | | | | |
| | 250k bit/s | <input checked="" type="checkbox"/> | | | | | | |
| DeviceNet Communication Data | Device Network Behavior Check All That Apply | Group 2 Client | <input type="checkbox"/> | Group 2 Only Client | <input type="checkbox"/> | | | |
| | | Group 2 Server | <input type="checkbox"/> | Group 2 Only Server | <input checked="" type="checkbox"/> | | | |
| | | Peer-to-Peer | <input type="checkbox"/> | Tool (not a device) | <input type="checkbox"/> | | | |
| | UCMM Explicit Message Groups Supported | Group 1 | <input type="checkbox"/> | Group 2 | <input type="checkbox"/> | Group 3 | <input type="checkbox"/> | |
| | | Group 1 | <input type="checkbox"/> | Group 2 | <input type="checkbox"/> | Group 3 | <input type="checkbox"/> | |
| | Dynamic I/O Message Groups Typical I/O Data Address Path | Input: Class | <u>4</u> | Inst. | <u>1</u> | Attr. | <u>3</u> | |
| | | Output: Class | <u>4</u> | Inst. | <u>2</u> | Attr. | <u>3</u> | |
| | Fragmented Explicit Messaging If yes, Acknowledge Time Out Typical Target Addresses (decimal) | Yes | <input checked="" type="checkbox"/> | | | | | |
| | | No | <input type="checkbox"/> | | | | | |
| | | Consumption | Service | <u>16</u> | Class | <u>1</u> | Inst. | <u>1</u> |
| Production | Service | <u>16</u> | Class | <u>1</u> | Inst. | <u>1</u> | Attr. | <u>7</u> |

| DeviceNet | | Identity Object 0x01 | | | | | | |
|---|---|----------------------|---|-------------------------------|--------------------------|--------------------------|---------------------|--|
| Required Object Implementation | Object Class | | ID | Description | Get | Set | Value Limits | |
| | Attributes | Open | 1 | Revision | x | <input type="checkbox"/> | _____ | |
| | | | 2 | Max instance | <input type="checkbox"/> | <input type="checkbox"/> | _____ | |
| | <input type="checkbox"/> None Supported | | 6 | Max ID of class attributes | <input type="checkbox"/> | <input type="checkbox"/> | _____ | |
| | | | 7 | Max ID of instance attributes | <input type="checkbox"/> | <input type="checkbox"/> | _____ | |
| | | | DeviceNet Services | | Parameter Options | | | |
| Services | | | <input type="checkbox"/> | Get_Attribute_All | _____ | | | |
| | | | <input type="checkbox"/> | Reset | _____ | | | |
| <input type="checkbox"/> None Supported | | | x | Get_Attribute_Single | _____ | | | |
| | | | <input type="checkbox"/> | Find_Next_Object_Instance | _____ | | | |
| Object Instance | Object Class | | ID | Description | Get | Set | Value Limits | |
| | Attributes | Open | 1 | Vendor | x | <input type="checkbox"/> | <u>89</u> | |
| | | | 2 | Product type | x | <input type="checkbox"/> | <u>0</u> | |
| | <input type="checkbox"/> None Supported | | 3 | Product code | x | <input type="checkbox"/> | <u>0</u> | |
| | | | 4 | Revision | x | <input type="checkbox"/> | <u>1</u> | |
| | | | 5 | Status | x | <input type="checkbox"/> | _____ | |
| | | | 6 | Serial number | x | <input type="checkbox"/> | _____ | |
| | | | 7 | Product name | x | <input type="checkbox"/> | <u>DeltaMax</u> | |
| | | | 8 | State | <input type="checkbox"/> | <input type="checkbox"/> | _____ | |
| | | | | DeviceNet Services | | Parameter Options | | |
| | Services | | | <input type="checkbox"/> | Get_Attribute_All | _____ | | |
| | <input type="checkbox"/> None Supported | | | x | Reset | <u>0</u> | | |
| | | | | x | Get_Attribute_Single | _____ | | |
| Vendor Specific Additions | | | If yes, fill out the Vendor Specific Additions form on page F- 7. | | Yes | <input type="checkbox"/> | | |
| | | | | | No | x | | |

| DeviceNet | | Message Router Object 0x02 | | | | | |
|---------------------------------------|---------------------------|----------------------------|---|-------------------------------|--------------------------|--------------------------|---------------------|
| Required Object Implementation | Object Class | | ID | Description | Get | Set | Value Limits |
| | Attributes | Open | 1 | Revision | <input type="checkbox"/> | <input type="checkbox"/> | ___ |
| | | | 4 | Optional attribute list | <input type="checkbox"/> | <input type="checkbox"/> | ___ |
| | X None Supported | | 5 | Optional service list | <input type="checkbox"/> | <input type="checkbox"/> | ___ |
| | | | 6 | Max ID of class attributes | <input type="checkbox"/> | <input type="checkbox"/> | ___ |
| | | | 7 | Max ID of instance attributes | <input type="checkbox"/> | <input type="checkbox"/> | ___ |
| | | | | | | | |
| | | | DeviceNet Services | | Parameter Options | | |
| Services | | | <input type="checkbox"/> | Get_Attribute_All | ___ | | |
| | | | <input type="checkbox"/> | Get_Attribute_Single | ___ | | |
| X None Supported | | | | | | | |
| | | | | | | | |
| | Object Instance | | ID | Description | Get | Set | Value Limits |
| | Attributes | Open | 1 | Object list | <input type="checkbox"/> | <input type="checkbox"/> | ___ |
| | | | 2 | Maximum connections supported | <input type="checkbox"/> | <input type="checkbox"/> | ___ |
| | X None Supported | | 3 | Number of active connections | <input type="checkbox"/> | <input type="checkbox"/> | ___ |
| | | | 4 | Active connections list | <input type="checkbox"/> | <input type="checkbox"/> | ___ |
| | | | | | | | |
| | | | DeviceNet Services | | Parameter Options | | |
| Services | | | <input type="checkbox"/> | Get_Attribute_All | ___ | | |
| X None Supported | | | <input type="checkbox"/> | Get_Attribute_Single | ___ | | |
| | | | | | | | |
| | Vendor Specific Additions | | If yes, fill out the Vendor Specific Additions form on page F- 7. | | Yes | <input type="checkbox"/> | |
| | | | | | No | x | |

| DeviceNet | | DeviceNet Object 0x03 | | | | |
|---|-------------------------------------|---|-----------------------------|--------------------------|--------------------------|---------------------|
| Required | Object Class | ID | Description | Get | Set | Value Limits |
| Object Implementation | Attributes Open X None Supported | 1 | Revision | <input type="checkbox"/> | <input type="checkbox"/> | ___ |
| | | DeviceNet Services | | Parameter Options | | |
| Services X None Supported | | <input type="checkbox"/> | Get_Attribute_Single | ___ | | |
| Object Instance | ID | Description | Get | Set | Value Limits | |
| Attributes Open | 1 | MAC ID | x | x | 0 to 63 | |
| | 2 | Baud rate | x | x | 0 to 2 | |
| <input type="checkbox"/> None Supported | 3 | BOI | x | <input type="checkbox"/> | ___ | |
| | 4 | Bus-off counter | x | x | 0 | |
| | 5 | Allocation information | x | <input type="checkbox"/> | ___ | |
| | 6 | MAC ID switch changed | x | <input type="checkbox"/> | ___ | |
| | 7 | Baud rate switch changed | <input type="checkbox"/> | <input type="checkbox"/> | ___ | |
| | 8 | MAC ID switch value | <input type="checkbox"/> | <input type="checkbox"/> | ___ | |
| | 9 | Baud rate switch value | <input type="checkbox"/> | <input type="checkbox"/> | ___ | |
| | | DeviceNet Services | | Parameter Options | | |
| Services | | x | Get_Attribute_Single | ___ | | |
| | | x | Set_Attribute_Single | ___ | | |
| <input type="checkbox"/> None Supported | | x | Allocate M/S connection set | ___ | | |
| | | x | Release M/S connection set | ___ | | |
| Vendor Specific Additions | | If yes, fill out the Vendor Specific Additions form on page F- 7. | | Yes | <input type="checkbox"/> | |
| | | | | No | x | |

| DeviceNet | Connection Object 0x05 | | | | | | |
|-----------------------|-----------------------------------|------|---------------------------|---------------------------|--------------------------|--------------------------|--------------|
| Required | Object Class | | ID | Description | Get | Set | Value Limits |
| Object Implementation | Attributes | Open | 1 | Revision | <input type="checkbox"/> | <input type="checkbox"/> | — |
| | X None Supported | | | | | | |
| | | | DeviceNet Services | | Parameter Options | | |
| | Services | | <input type="checkbox"/> | Reset | — | | |
| | | | <input type="checkbox"/> | Create | — | | |
| | X None Supported | | <input type="checkbox"/> | Delete | — | | |
| | | | <input type="checkbox"/> | Get_Attribute_Single | — | | |
| | | | <input type="checkbox"/> | Find_Next_Object_Instance | — | | |
| | Total Active Connections Possible | | <u>3</u> | | | | |

| Object Instance 1 | Section | Information | Max Instance |
|---|--------------------|--|--------------|
| <i>The Object Instance section must be completed for each combination of Instance type, Production trigger, Transport type, and Transport class supported</i> | Instance type | Explicit Message x Polled I/O <input type="checkbox"/> Bit Strobed I/O <input type="checkbox"/> Dynamic I/O <input type="checkbox"/> | <u>1</u> |
| | Production trigger | Cyclic <input type="checkbox"/> Change of State <input type="checkbox"/> Application Triggered <input type="checkbox"/> | |
| | Transport type | Server <input type="checkbox"/> Client <input type="checkbox"/> | |
| | Transport class | 0 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> | |

| | | ID | Description | Get | Set | Value Limits | |
|------------|------|----|---------------------------------|----------|--------------------------|--------------|--|
| Attributes | Open | 1 | State | x | <input type="checkbox"/> | — | |
| | | 2 | Instance type | x | <input type="checkbox"/> | — | |
| | | 3 | Transport class trigger | x | <input type="checkbox"/> | — | |
| | | 4 | Produced connection ID | x | <input type="checkbox"/> | — | |
| | | 5 | Consumed connection ID | x | <input type="checkbox"/> | — | |
| | | 6 | Initial comm. characteristics | x | <input type="checkbox"/> | — | |
| | | 7 | Produced connection size | x | <input type="checkbox"/> | — | |
| | | 8 | Consumed connection size | x | <input type="checkbox"/> | — | |
| | | 9 | Expected packet rate | x | x | — | |
| | | 12 | Watchdog time-out action | x | <input type="checkbox"/> | — | |
| | | 13 | Produced connection path length | x | <input type="checkbox"/> | — | |
| | | 14 | Produced connection path length | x | <input type="checkbox"/> | — | |
| | | 15 | Consumed connection path length | x | <input type="checkbox"/> | — | |
| | | 16 | Consumed connection path length | x | <input type="checkbox"/> | — | |
| | | | Consumed connection path | | | | |

| | DeviceNet Services | Parameter Options | |
|---------------------------|---|---|-------------------------------------|
| Services | <input checked="" type="checkbox"/> Reset <input type="checkbox"/> Delete <input type="checkbox"/> Apply_Attributes <input checked="" type="checkbox"/> Get_Attribute_Single <input checked="" type="checkbox"/> Set_Attribute_Single | _____ _____ _____ _____ _____ | |
| Vendor Specific Additions | If yes, fill out the Vendor Specific Additions form on page F- 7. | Yes | <input type="checkbox"/> |
| | | No | <input checked="" type="checkbox"/> |

Connection Object 0x05 (Continued)

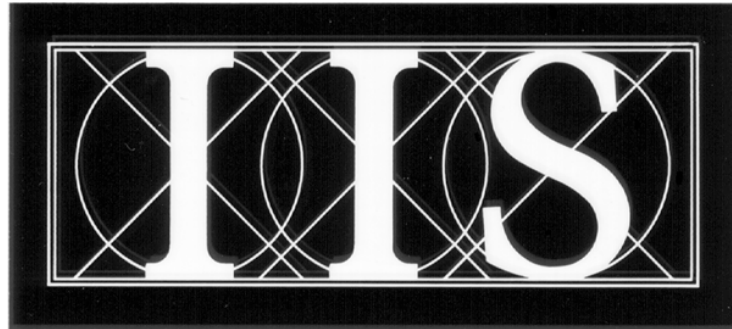
| Object Instance 2 | Section | Information | Max Instance |
|---|---|--|--|
| <i>The Object Instance section must be completed for each combination of Instance type, Production trigger, Transport type, and Transport class supported</i> | Instance type | Explicit Message <input type="checkbox"/> Polled I/O x Bit Strobed I/O <input type="checkbox"/> Dynamic I/O <input type="checkbox"/> | <u>1</u> |
| | Production trigger | Cyclic Change of State Application Triggered | <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> |
| | Transport type | Server Client | <input type="checkbox"/> <input type="checkbox"/> |
| | Transport class | 0 2 3 | <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> |
| | | | |
| Attributes | Open | ID Description | Get Set Value Limits |
| | | 1 State | x <input type="checkbox"/> — |
| | | 2 Instance type | x <input type="checkbox"/> — |
| | | 3 Transport class trigger | x <input type="checkbox"/> — |
| | | 4 Produced connection ID | x <input type="checkbox"/> — |
| | | 5 Consumed connection ID | x <input type="checkbox"/> — |
| | | 6 Initial comm. characteristics | x <input type="checkbox"/> — |
| | | 7 Produced connection size | x <input type="checkbox"/> — |
| | | 8 Consumed connection size | x <input type="checkbox"/> — |
| | | 9 Expected packet rate | x x — |
| | | 12 Watchdog time-out action | x <input type="checkbox"/> — |
| | | 13 Produced connection path | x <input type="checkbox"/> — |
| | | 14 length | x <input type="checkbox"/> — |
| | | 15 Produced connection path | x <input type="checkbox"/> — |
| | | 16 Consumed connection path | x <input type="checkbox"/> — |
| | | Consumed connection path | |
| | | Services | DeviceNet Services |
| x | Reset | | — |
| <input type="checkbox"/> | Delete | | — |
| <input type="checkbox"/> | Apply_Attributes | | — |
| x | Get_Attribute_Single | | — |
| x | Set_Attribute_Single | — | |
| Vendor Specific Additions | If yes, fill out the Vendor Specific Additions form on page F- 7. | Yes <input type="checkbox"/> No x | |

Connection Object 0x05 (Continued)

| Object Instance 3 | Section | Information | Max Instance | | |
|---|---|--|--|-------------------------------------|--------------------------|
| <i>The Object Instance section must be completed for each combination of Instance type, Production trigger, Transport type, and Transport class supported</i> | Instance type | Explicit Message <input type="checkbox"/> Polled I/O <input type="checkbox"/> Bit Strobed I/O x Dynamic I/O <input type="checkbox"/> | 1 | | |
| | Production trigger | Cyclic Change of State Application Triggered | <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> | | |
| | Transport type | Server Client | <input type="checkbox"/> <input type="checkbox"/> | | |
| | Transport class | 0 2 3 | <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> | | |
| | Attributes | Open | ID Description | Get Set Value Limits | |
| | | | 1 State | x <input type="checkbox"/> — | |
| | | | 2 Instance type | x <input type="checkbox"/> — | |
| | | | 3 Transport class trigger | x <input type="checkbox"/> — | |
| | | | 4 Produced connection ID | x <input type="checkbox"/> — | |
| | | | 5 Consumed connection ID | x <input type="checkbox"/> — | |
| 6 Initial comm. characteristics | | | x <input type="checkbox"/> — | | |
| 7 Produced connection size | | | x <input type="checkbox"/> — | | |
| 8 Consumed connection size | | | x <input type="checkbox"/> — | | |
| 9 Expected packet rate | | | x x — | | |
| 12 Watchdog time-out action | | | x <input type="checkbox"/> — | | |
| 13 Produced connection path length | | | x <input type="checkbox"/> — | | |
| 14 Produced connection path length | | | x <input type="checkbox"/> — | | |
| 15 Consumed connection path length | | | x <input type="checkbox"/> — | | |
| 16 Consumed connection path length | | | x <input type="checkbox"/> — | | |
| | | | Consumed connection path | | |
| Services | | | DeviceNet Services | | Parameter Options |
| | x | Reset | — | | |
| | <input type="checkbox"/> | Delete | — | | |
| | <input type="checkbox"/> | Apply_Attributes | — | | |
| | x | Get_Attribute_Single | — | | |
| | x | Set_Attribute_Single | — | | |
| Vendor Specific Additions | If yes, fill out the Vendor Specific Additions form on page F- 7. | Yes <input type="checkbox"/> No x | | | |

| | | | | | | |
|-----------------------------------|---|---|------------------------------|---------------------------|--------------------------|----------------------------|
| DeviceNet | OBJECT NAME <u>Assembly Object</u> | | OBJECT ID <u>0x04</u> | | | |
| Open Object Implementation | Object Class | ID | Description | Get | Set | Value Limits |
| | Attributes | Open | 1 | Revision | <input type="checkbox"/> | <input type="checkbox"/> 1 |
| | X None Supported | | 2 | Max instance | <input type="checkbox"/> | <input type="checkbox"/> — |
| | | | DeviceNet Services | | Parameter Options | |
| | Services | | <input type="checkbox"/> | Create | — | |
| | X None Supported | | <input type="checkbox"/> | Delete | — | |
| | | | x | Get_Attribute_Single | — | |
| | Object Instance | Section | Information | Max Instance | | |
| | <i>The Object Instance section must be completed for each combination of Instance type.</i> | Instance type | Static Input | <input type="checkbox"/> | | |
| | <input type="checkbox"/> None Supported | | Static Output | <input type="checkbox"/> | | |
| | | | Static I/O | x | | |
| | | | Static Configuration | <input type="checkbox"/> | | |
| | | ID | Description | Get | Set | Value Limits |
| | Attributes | Open | 1 | Number of members in list | <input type="checkbox"/> | <input type="checkbox"/> — |
| | <input type="checkbox"/> None Supported | | 2 | Member list | <input type="checkbox"/> | <input type="checkbox"/> — |
| | | | 3 | Data | x | x — |
| | | | DeviceNet Services | | Parameter Options | |
| | Services | | <input type="checkbox"/> | Delete | — | |
| | <input type="checkbox"/> None Supported | | x | Get_Attribute_Single | — | |
| | | | x | Set_Attribute_Single | — | |
| | | | <input type="checkbox"/> | Add_Member | — | |
| | | | <input type="checkbox"/> | Delete_Member | — | |
| | Vendor Specific Additions | If yes, fill out the Vendor Specific Additions form on page F- 7. | | Yes | <input type="checkbox"/> | |
| | | | | No | x | |

IB-19B005



**INDUSTRIAL
INDEXING SYSTEMS
INC.**

**626 FISHERS RUN
VICTOR, NEW YORK 14564**

**(585) 924-9181
FAX: (585) 924-2169**

PRINTED IN USA
© 1998