acontis technologies GmbH

SOFTWARE

EC-Master

EtherCAT Master Stack
for embedded Operating Systems

Master Object Dictionary

Edition: 2023-11-30

Content - compact

1 Introduction

1.1 What is Master Object Dictionary?

Like slaves may support diagnosis and configuration that can be accessed by means of asynchronous SDO uploads, the EC-Master stack contains the possibility to get access to such objects similarly. The master implements the mandatory objects of the slave and enhances some useful things in the vendor definable areas.

Find a close documentation to Object Dictionary structure in this document.

1.2 How to access Master Object dictionary

To let an application have access to the Master's Object dictionary, a special "Slave" is simulated inside the Master. This simulated slave does not support any slave operations despite the asynchronous CoE access and the CoE Information Services.

To access the object dictionary the Slave ID *MASTER_SLAVE_ID* can be used for access of:
emCoeSdoUpload(…,MASTER_SLAVE_ID, …);
emCoeSdoDownload(…,MASTER_SLAVE_ID, …);
emCoeGetODList(…,MASTER_SLAVE_ID, …);
…

Some objects, e.g. 0x1018 are read-only for emCoeSdo… access, but should be forced configurable by the application. This can be accomplished by means of `EC_IOCTL_MASTEROD_SET_VALUE`, see below.
Complete access to objects of type OBJCODE_VAR is not possible.

The master object dictionary follows the little-endian byte ordering convention.

1.3 ecatIoControl – EC_IOCTL_MASTEROD_SET_VALUE

This IOCTL forces the content change of the entry with given index and sub-index. See object overview below to know which objects can be forced using EC_IOCTL_MASTEROD_SET_VALUE.

Parameters

pbyInBuf

[in] Pointer to variable of type EC_T_MASTEROD_OBJECT_PARMS.

dwInBufSize

[in] Size of the input buffer provided at *pbyInBuf* in bytes.

pbyOutBuf

[] Should be set to EC_NULL.

dwOutBufSize

[] Should be set to 0.

pdwNumOutData

[] Should be set to EC_NULL.

Comment

The values are kept in case of ecatMasterConfigure(…). Complete access is not supported.

```
typedef struct _EC_T_MASTEROD_OBJECT_PARMS
{
   EC_T_WORD          wIndex;
   EC_T_BYTE          bySubindex;
   EC_T_BYTE          byReserved;
   EC_T_BYTE*         pbyData;
   EC_T_DWORD         dwLength;
   EC_T_DWORD         dwReserved;
} EC_PACKED(1) EC_T_MASTEROD_OBJECT_PARMS;
```

Description

wIndex

[in] Object's index, e.g. 0x1018

bySubindex

[in] Object's sub-index, e.g. 1

pbyData

[in] Pointer to object's data to be written

dwLength

[in] Data length to be written

dwLength is the data’s size in bytes. Add +1 to the length of zero-terminated strings for termination.

2 Object Description

2.1 Overview

Communication Area Objects: 0x1000-0x1FFF

Index	Object Type	Name	Type	M/O/C[1]	Access A / F / C[2]
0x1000	VAR	Device Type	Unsigned32	M	A
0x1001	VAR	Error Register	Unsigned8	O	-
0x1008	VAR	Manufacturer Device Name String	VisibleString	O	A, F
0x1009	VAR	Manufacturer Hardware Version String	VisibleString	O	A, F
0x100A	VAR	Manufacturer Software Version String	VisibleString	O	A, F
0x1018	RECORD	Identity Object	Identity (0x23)	M	A, F, C
0x10F3	RECORD	History Object	History (0x26)	O	A

Generic Master Objects: 0x2000-0x20FF

Index	Object Type	Name	Type	M/O/C	Access
0x2000	VAR	Master State Change Command Register	Unsigned32	C	A
0x2001	VAR	Master State Summary	Unsigned32	C	A
0x2002	RECORD	Bus Diagnosis Object	BusDiagnostic (0x40)	C	A, C
0x2003	RECORD	Redundancy Object	Redundancy (0x52)	O	A
0x2004	RECORD	Notify Counter Object	Notify Counter (0x53)	O	A
0x2005	RECORD	MAC Address	MACAddress (0x41)	C	A, C
0x2010	VAR	Debug Register	Unsigned64	C	A
0x2020	RECORD	Master Init. Parameters	MasterInitParm (0x42)	C	A, C

Distributed Clocks Objects: 0x2100-0x21FF

Index	Object Type	Name	Type	M/O/C	Access
0x2100	VAR	DC Slave Sync Deviation Limit	Unsigned32	C	A
0x2101	VAR	DC Current Deviation	Signed32	C	A
0x2102	RECORD	DCM Bus Shift	Bus Load Base (0x44)	C	-
0x2103	VAR	DC Reserved	Unsigned32	C	-

[1] Mandatory, Optional, Customer

[2] A: Availble; F: Force, See `EC_IOCTL_MASTEROD_SET_VALUE` below. ; C: Complete Access

Slave Configuration / Information Objects: 0x3000-0x3FFF

0x3000 - 0x3FFF	RECORD	Slave Configuration and Information Objects	SlaveCfgInfo (0x43)	C	A,C

Slave Configuration Objects: 0x8000-0x8FFF

0x8000 - 0x8FFF	RECORD	One index entry for each configured slave (from ENI)	SlaveCfg (0x45)	M	A,C

Slave Information Objects: 0x9000-0x9FFF

0x9000 - 0x9FFF	RECORD	One index entry for each connected slave (updated during bus scan)	SlaveInfo (0x46)	O	A,C

Slave Diagnosis Data Objects: 0xA000-0xAFFF

0xA000 - 0xAFFF	RECORD	One subindex entry for each connected slave (cyclic updated)	SlaveDiag (0x47)	O	A,C

Device Area Objects: 0xF000-0xFFFF

0xF000	RECORD	Modular Device Profile	DeviceProfile (0x48)	M	A, C
0xF002	RECORD	Detect Modules Command	DetectModCmd (0x49)	M	A
0xF020 – 0xF02F	RECORD	Configured Address List	ConfAddrList (0x50)	M	A, C
0xF040 – 0xF04F	RECORD	Detected Address List	ConnAddrList (0x51)	O	A, C
0xF120	RECORD	Frame Statistics	FrameStatistics (0x56)	M	A
0xF200	RECORD	Diag Interface Control	DiagInterfaceCtl (0x57)	M	A

2.2 Communication Area Objects: 0x1000-0x1FFF

Index	Object Type	Name	Type	M/O/C	Available
0x1000	VAR	Device Type	Unsigned32	M	X
0x1001	VAR	Error Register	Unsigned8	O	-
0x1008	VAR	Manufacturer Device Name String	VisibleString	O	X
0x1009	VAR	Manufacturer Hardware Version String	VisibleString	O	X
0x100A	VAR	Manufacturer Software Version String	VisibleString	O	X
0x1018	RECORD	Identity Object	Identity (0x23)	M	X
0x10F3	RECORD	History Object	History (0x26)	O	X

2.2.1 Device Type 0x1000

Attribute	Value
Name	Device Type
Object Code	VAR
Data Type	Unsigned32
Category	Mandatory
Access	Ro
Value	1100

2.2.2 Manufacturer Device Name String 0x1008

Attribute	Value
Name	Device Type
Object Code	VAR
Data Type	VisibleString
Category	Optional
Access	Ro
Value	“EC-Master”

Log:

```
051217(00000041): 1008 “  Device name”: type 0x0009, code=0x07, optional, SubIds=0
051219(00000041): 1008:0 “  Device name” => type=0x0009, len=88, R .R .R
051219(00000041): 1008:0 STRG : EC-Master
```

2.2.3 Manufacturer Hardware Version String 0x1009

Attribute	Value
Name	Device Type
Object Code	VAR
Data Type	VisibleString
Category	Optional
Access	Ro

Value	“V MM.mm.ss.bb”
	MM = Major Version
	mm = Minor Version
	ss = Service Pack
	bb = Build
	e.g. “V 01.05.02.02”

Log:

```
1009 ‟          Hardware version”: type 0x0009, code=0x07, optional, SubIds=0
1009:0 ‟          Hardware version” => type=0x0009, len=112, R .R .R
1009:0 STRG : V 01.05.02.02
```

2.2.4 Manufacturer Software Version String 0x100A

Attribute	Value
Name	Device Type
Object Code	VAR
Data Type	VisibleString
Category	Optional
Access	Ro

Value	“V MM.mm.ss.bb”
	MM = Major Version
	mm = Minor Version
	ss = Service Pack
	bb = Build
	e.g. “V 01.05.02.02”

Log:

```
100a ‟         Software version”: type 0x0009, code=0x07, optional, SubIds=0
100a:0 ‟         Software version” => type=0x0009, len=112, R .R .R
100a:0 STRG : V 01.05.02.02
```

2.2.5 Identity Object 0x1018

Subindex	Description	Data Type	Access	M/O/C	Available
0	Number of Entries	Unsigned8	R	M	X
1	Vendor ID	Unsigned32	R	M	X
2	Product Code	Unsigned32	R	M	X
3	Revision Number EC Master SW Rev.	Unsigned32	R	O	X
4	Serial Number	Unsigned32	R	O	X

Datatype for application access:

```
typedef struct _EC_T_OBJ1018{
       EC_T_WORD   wSubIndex0;        /* = 4 */
       EC_T_DWORD  dwVendorID;        /* = 0x00004154 */
       EC_T_DWORD  dwProductcode;     /* = 0x6d657461 */
       EC_T_DWORD  dwRevision;        /* SW Revision e.g. 0x01050202 */
       EC_T_DWORD  dwSerialnumber;    /* = 0 */
} EC_PACKED(1) EC_T_OBJ1018;
```

Log :

```
1018 «                    Identity » : type 0x0023, code=0x09, optional, SubIds=4

1018 :0 «                SubIndex 000 » => type=0x0005, len=08, R .R .R

1018 :0 BYTE : 0x04

1018 :1 «                   Vendor ID » => type=0x0007, len=32, R .R .R

1018 :1 DWRD : 0x00004154

1018 :2 «                Product code » => type=0x0007, len=32, R .R .R

1018 :2 DWRD : 0x6D657461

1018 :3 «                    Revision » => type=0x0007, len=32, R .R .R

1018 :3 DWRD : 0x01050202

1018 :4 «               Serial number » => type=0x0007, len=32, R .R .R

1018 :4 DWRD : 0x00000000
```

2.2.6 History Object 0x10F3

Subindex	Description	Data Type	Access	M/O/C	Available
0	Number of Entries	Unsigned8	R	M	X
1	Maximum number of Diag messages	Unsigned8	R	M	X
2	Subindex of newest Diag message	Unsigned8	R	M	X
3	Subindex of newest acknowledged Diag message	Unsigned8	RW	M	X
4	New Diag messages available	BOOL32	R	M	X

5	Flags
	`Flags (UINT16, RW)`
	`Bit 0 = 1: (R only)`
	`Enable Emergency sending (default = 0)`
	`Bit 1 = 1:`
	`Disable Storing Info Messages (default = 0)`
	`Bit 2 = 1:`
	`Disable Storing Warning Messages (default = 0)`
	`Bit 3 = 1:`
	`Disable Storing Error Messages (default = 0)`
	`Bit 4-15:`
	`reserved for future use`
		Unsigned16	RW	M	-

6-255	Diagnosis messages	See below	R	M	X

Datatype for application access:

```
typedef struct _EC_T_OBJ10F3 {
       EC_T_WORD              wSubIndex0;
       EC_T_BYTE              byMaxDiagMessages;
       EC_T_BYTE              bySubIndexMessageNew;
       EC_T_BYTE              bySubIndexMessageAck;
       EC_T_BOOL              bNewDiagMessages;
       EC_T_WORD              wFlags;
       EC_T_OBJ10F3_DIAGMSG   aoDiagMessages[249];    /* data appended */
} EC_PACKED(1) EC_T_OBJ10F3;
```

2.2.6.1 Diagnosis Messages Object 0x10F3: Subindex 6-255

Byte-Offset	Description	Data Type
0	Diag-Number Bit 0-11: free use Bit 12-15 = 14: to be comp. with Emergency Error Bit 16-31 = 0: reserved Bit 16-31 = 1-0xFFFE: free use Bit 16-31 = 0xFFFF: reserved	Unsigned32
4	Flags Bit 0-3: Diag type (0= Info, 1 = warning, 2 = error) Bit 4-15: reserved	Unsigned16
6	Text ID 0 = no Text ID 1-65535 = Reference to a Text ID with formatted string	Unsigned16
8	Time Stamp in ns (from DC)	Unsigned64
16	Flags parameter 1	Unsigned16
18	Parameter 1	several
N	Flags parameter n	Unsigned16
N+2	Parameter n	several

Datatype for application access:

```
typedef struct _EC_T_OBJ10F3_DIAGMSG {
        EC_T_DWORD  dwDiagNumber;
        EC_T_WORD   wFlags;
        EC_T_WORD   wTextId;
        EC_T_UINT64 qwTimeStamp;
        EC_T_BYTE   byParams[HISTORY_OBJECT_DIAGELE_SIZE];
} EC_PACKED(1) EC_T_OBJ10F3_DIAGMSG;
```

2.3 Generic Master Objects: 0x2000-0x20FF

Index	Object Type	Name	Type	M/O/C	Available
0x2000	VAR	Master State Change Command Register	Unsigned32	C	X
0x2001	VAR	Master State Summary	Unsigned32	C	X
0x2002	RECORD	Bus Diagnosis Object	BusDiagnostic (0x40)	C	X
0x2005	RECORD	MAC Address	MACAddress (0x41)	C	X
0x2010	VAR	Debug Register	Unsigned48	C	X
0x2020	RECORD	Master Init. Parameters	MasterInitParm (0x42)	C	X

2.3.1 Master State Change Command Register 0x2000

Attribute	Value
Name	Master State
Object Code	VAR
Data Type	Unsigned32
Category	Customer
Access	RW
Value	0 = invalid 1 = init 2 = pre-operational 3 = bootstrap mode 4 = safe operational 8 = operational

2.3.2 Master State Summary 0x2001

Attribute	Value
Name	Master State
Object Code	VAR
Data Type	Unsigned32
Category	Customer
Access	R
Value	Bit 0: = 1 Master o.k. Bit 1-3: Reserved Bit 4-7: Master State Bit 8: Slaves in requested State Bit 9: Master in requested State Bit 10: Bus Scan Match Bit 11: Reserved Bit 12: DC is enabled Bit 13: DC In-Sync Bit 14: DC Busy Bit 15: Reserved Bit 16: Link Up Bit 17-31: Reserved

The parameter data itself (Offset 16ff) consists of a flag field and a bunch of data, defined through flag field:

Parameter Flags Bit 12-15	Parameter Flags Bit 0-11	Type of Data	Data
0	CoE DataType e.g. 0x0007 = UINT32	Data Type	Data defined through CoE DataType
1	Length in Byte	Byte Array	Byte stream byData[Size]
2	Length in Byte	ASCII-String	String szString[Length] (not '\0' terminated)
3	Length in Byte	Unicode String	String wszString[Length/2] (not L'\0' terminated)
4	0	Text Id	Text Id (Word)

2.3.3 Bus Diagnosis Object 0x2002

Object Type: RECORD, Manufacturer Specific Identity 0x40

Subindex	Description	Data Type	Access	M/O/C	Availa ble
0	Number of Entries	Unsigned8	R	C	X
1	Reserved	Unsigned16	R	C	X
2	Configuration Checksum CRC32	Unsigned32	R	C	X
3	Number of found Slaves	Unsigned32	R	C	X
4	Number of found DC Slaves	Unsigned32	R	C	X
5	Number of Slaves in Configuration	Unsigned32	R	C	X
6	Number of Mailbox slaves in Config.	Unsigned 32	R	C	X
7	Counter: TX frames	Unsigned32	R	C	X
8	Counter: RX frames	Unsigned32	R	C	X
9	Counter: Lost frames	Unsigned32	R	C	X
10	Counter: Cyclic frames	Unsigned32	R	C	X
11	Counter: Cyclic datagrams	Unsigned32	R	C	X
12	Counter: Acyclic frames	Unsigned32	R	C	X
13	Counter: Acyclic datagrams	Unsigned32	R	C	X
14	Clear Counters by writing 1 to bit(s) Bit 0: Clear all Counters Bit 1: Clear Tx Frame Counter (Idx 7) Bit 2: Clear Rx Frame Counter (Idx 8) Bit 3: Clear Lost Frame Counter (Idx 9) Bit 4: Clear Cyclic Frame Counter (Idx 10) Bit 5: Clear Cyclic Datagram Counter (Idx 11) Bit 6: Clear Acyclic Frame Counter (Idx 12) Bit 7: Clear Acyclic DataGram Counter (Idx 13) Bit 7-31: Reserved	Unsigned32	RW	C	X

Datatype for application access:

```
typedef struct _EC_T_OBJ2002 {
        EC_T_WORD    wSubIndex0;
        EC_T_WORD    wReserved;
        EC_T_DWORD   dwCRC32ConfigCheckSum;
        EC_T_DWORD   dwNumSlavesFound;
        EC_T_DWORD   dwNumDCSlavesFound;
        EC_T_DWORD   dwNumCfgSlaves;
        EC_T_DWORD   dwNumMbxSlaves;
        EC_T_DWORD   dwTXFrames;
        EC_T_DWORD   dwRXFrames;
        EC_T_DWORD   dwLostFrames;
        EC_T_DWORD   dwCyclicFrames;
        EC_T_DWORD   dwCyclicDatagrams;
        EC_T_DWORD   dwAcyclicFrames;
        EC_T_DWORD   dwAcyclicDatagrams;
        EC_T_DWORD   dwClearCounters;
} EC_PACKED(1) EC_T_OBJ2002;
```

2.3.4 Redundancy Diagnosis Object 0x2003

Object Type: RECORD, Manufacturer Specific Identity 0x52

Subindex	Description	Data Type	Access	M/O/C	Availa ble
0	Number of Entries	Unsigned8	R	C	X
1	Red Enabled	Unsigned8	R	C	X
2	Main Line Slave Count	Unsigned16	R	C	X
3	Red Line Slave Count	Unsigned16	R	C	X
4	Line Break Detected	Unsigned8	R	C	X

Datatype for application access:

```
typedef struct _EC_T_OBJ2003 {
       EC_T_WORD   wSubIndex0;
       EC_T_BYTE   byRedEnabled;
       EC_T_DWORD  wNumOfMainSlaves;
       EC_T_DWORD  wNumOfRedSlaves;
       EC_T_BYTE   byLineBreak;
} EC_PACKED(1) EC_T_OBJ2003;
```

2.3.5 *Notify Counter Object 0x2004*

Object Type: RECORD, Manufacturer Specific Identity 0x53

Subindex	Description	Data Type	Access	M/O/C	Available
0	Number of Entries	Unsigned8	R	C	X
1	Maximum Messages	Unsigned8	R	C	X
2	Message Count	Unsigned8	R	C	X
3	Reserved	Unsigned8	-	-	-
4 …	Messages	OBJ2004_NOTIFYMSG	R	C	X

Datatype for application access:

```
typedef struct _EC_T_OBJ2004 {
        EC_T_WORD    wSubIndex0;
        EC_T_BYTE    byMaxMessages;
        EC_T_BYTE    byMessageCount;
        EC_T_BYTE    wNumOfRedSlaves;
        EC_T_BYTE    byReserved;

        /* SubIndex 004 and following see EC_T_OBJ2004_NOTIFYMSG below */
} EC_PACKED(1) EC_T_OBJ2004;
```

2.3.5.1 Notify Counter Object 0x2004: Subindex 4-255

Byte-Offset	Description	Data Type
0	Notification Code	Unsigned32
4	Notification Count	Unsigned32

Datatype for application access:

```
typedef struct _EC_T_OBJ2004_NOTIFYMSG {
        EC_T_DWORD  dwCode;
        EC_T_DWORD  dwCount;
} EC_PACKED(1) EC_T_OBJ2004_NOTIFYMSG;
```

2.3.6 *MAC Address 0x2005*

Object Type: RECORD, Manufacturer Specific Identity 0x41

Subindex	Description	Data Type	Access	M/O/C	Available
0	Number of Entries	Unsigned8	R	C	X
1	Hardware	Unsigned48	R	C	X
2	Red Hardware	Unsigned48	R	C	X
3	Configuration Source	Unsigned48	R	C	X
4	Configuration Destination	Unsigned48	R	C	X

Datatype for application access:

```
typedef struct _EC_T_OBJ2005 {
        EC_T_WORD  wSubIndex0;
        EC_T_BYTE  abyHardware[6];
        EC_T_BYTE  abyRedHardware[6];
        EC_T_BYTE  abyCfgSource[6];
        EC_T_BYTE  abyCfgDestination[6];
} EC_PACKED(1) EC_T_OBJ2005;
```

2.3.7 *Mailbox Statistics Object 0x2006*

Object Type: RECORD, Manufacturer Specific Identity 0x54

Subindex	Description	Data Type	Access	M/O/C	Available
0	Number of Entries	Unsigned8	R	C	X
1	AoE Total Read Transfer Count	Unsigned32	R	C	X
2	AoE Read Transfer Count Last Second	Unsigned32	R	C	X
3	AoE Total Bytes Read	Unsigned32	R	C	X
4	AoE Bytes Read Last Second	Unsigned32	R	C	X
5 … 8	AoE *Write* Statistics Total / Last Second	Unsigned32	R	C	X
9 … 16	*CoE* Read / Write Statistics Total / Last Second	Unsigned32	R	C	X
17…24	EoE	Unsigned32	R	C	X
25…32	FoE	Unsigned32	R	C	X
33…40	SoE	Unsigned32	R	C	X
41…48	VoE	Unsigned32	R	C	X
49…56	RawMbx	Unsigned32	R	C	X
57…64	Reserved	Unsigned32	R	C	X
65	Clear Counters by writing 1 to bit(s) Bit 0…63: Clear Counter at SubIndex n, with: Bit 0: n = 1, Bit 1: n = 2 …, Bit 63: n = 64 E.g. 0x0000000100: Clear CoE Total Read Transfer Count	Unsigned64	RW	C	X

Datatype for application access:

```
typedef struct _EC_T_OBJ2006 {
        EC_T_WORD       wSubIndex0;
        EC_T_DWORD      dwCnt[64];
        EC_T_UINT64     qwClearCounters;
} EC_PACKED(1) EC_T_OBJ2006;
```

2.3.8 Add History Diagnosis Message Command Object 0x2007

Object Type: RECORD, Manufacturer Specific Identity 0x58

Subindex	Description	Data Type	Access	M/O/C	Available
0	Number of Entries	Unsigned8	R/W	M	X
1	Diagnosis code	Unsigned32	W	M	X
2	Message type	Unsigned8	W	M	X
3	Message text ID	Unsigned16	W	M	X
4	Length of the parameters data	Unsigned8	W	M	X
5	Parameters data according to Object 0x10F3	…	W	M	X

Datatype for application access:

```
typedef struct _EC_T_OBJ2007_HISTORY_ADD_DIAGMSG
{
    EC_T_WORD               wSubIndex0;
    EC_T_DWORD              dwDiagCode;
    EC_T_BYTE               byType;
    EC_T_WORD               wTextID;
    EC_T_BYTE               byParmsLen;
    EC_T_BYTE               abyParms[HISTORY_OBJECT_DIAGELE_SIZE];
} EC_PACKED(1) EC_T_OBJ2007_HISTORY_ADD_DIAGMSG;
```

2.3.9 Debug Register 0x2010

Attribute	Value
Name	Debug Register
Object Code	VAR
Data Type	Unsigned64
Category	Customer
Access	RW
Value	Upper 16Bit: 0: activate LinkError Messages 1-15: reserved Lower 32Bit: Definition of parameter dwStateChangeDebug in structure EC_T_MASTER_CONFIG (Chapter 2.2.3.1 ecatInitMaster)

2.3.10 Master Init Parameters 0x2020

Object Type: RECORD, Manufacturer Specific Identity 0x42

Sub-index	Description	Data Type	Access	M/O/C	Available
0	Number of Entries	Unsigned8	R	C	X
1	EC_T_INITMASTERPARMS.dwVersion Application	Unsigned32	R	C	X
2	dwVersion Master	Unsigned32	R	C	
3	EC_T_MASTER_CONFIG.nSlaveMultiplier	Unsigned32	R	C	X
4	EC_T_MASTER_CONFIG.dwEcatCmdTimeout in millisec	Unsigned32	R	C	X
5	EC_T_MASTER_CONFIG.dwEcatCmdMaxRetries	Unsigned32	R	C	X
6	EC_T_MASTER_CONFIG.dwBusCycleTimeUsec in usec	Unsigned32	R	C	X
7	EC_T_MASTER_CONFIG.dwEoeTimeout in millisec	Unsigned32	R	C	X
8	EC_T_MASTER_CONFIG.dwFoeBusyTimeout in millisec	Unsigned32	R	C	X
9	EC_T_MASTER_CONFIG.dwMaxQueuedEthFrames	Unsigned32	R	C	X
10	EC_T_MASTER_CONFIG.dwMaxSlaveCmdPerFrame	Unsigned32	R	C	X
11	Reserved	Unsigned32	R	C	X
12	Reserved	Unsigned32	R	C	X
13	EC_T_MASTER_CONFIG.dwStateChangeDebug	Unsigned32	R	C	X
14	EC_T_LINK_DEV_PARAM.szDriverIdent	VisibleString	R	C	X
15	EC_T_LINK_DEV_PARAM.bPollingModeActive	Bool32	R	C	X
16	EC_T_LINK_DEV_PARAM.bAllocSendFrameActive	Bool32	R	C	X

Datatype for application access:

```
#define EC_OD_MAX_DRIVER_IDENT_LEN 40  /* maximum length of link layer name */

typedef struct _EC_T_OBJ2020 {
        EC_T_WORD   wSubIndex0;
        EC_T_DWORD  dwApplicationVersion;
        EC_T_DWORD  dwMasterVersion;
        EC_T_DWORD  dwSlaveMultiplier;
        EC_T_DWORD  dwEcatCmdTimeout;
        EC_T_DWORD  dwEcatCmdMaxRetries;
        EC_T_DWORD  dwBusCycleTimeUsec;
        EC_T_DWORD  dwEoeTimeout;
        EC_T_DWORD  dwFoeBusyTimeout;
        EC_T_DWORD  dwMaxQueuedEthFrames;
        EC_T_DWORD  dwMaxSlaveCmdPerFrame;
        EC_T_DWORD  dwReserved1;
        EC_T_DWORD  dwReserved2;
        EC_T_DWORD  dwStateChangeDebug;
        EC_T_CHAR   szDriverIdent[EC_OD_MAX_DRIVER_IDENT_LEN];
        EC_T_BOOL   bPollingModeActive;
        EC_T_BOOL   bAllocSendFrameActive;
} EC_PACKED(1) EC_T_OBJ2020;
```

2.4 Distributed Clocks Objects: 0x2100-0x21FF

Index	Object Type	Name	Type	M/O/C	Available
0x2100	VAR	DC Slave Sync Deviation Limit	Unsigned32	C	X
0x2101	VAR	DC Current Deviation	Signed32	C	X
0x2102	RECORD	DCM Bus Shift Object	Unsigned32	C	X
0x2103	VAR	DC Reserved	Unsigned32	C	-

2.4.1 Distributed Clocks Slave Sync Deviation Limit 0x2100

Attribute	Value
Name	Master State
Object Code	VAR
Data Type	Unsigned32
Category	Customer
Access	R
Value	dwDevLimit

2.4.2 Distributed Clocks Current Deviation 0x2101

Attribute	Value
Name	Master State
Object Code	VAR
Data Type	Signed32
Category	Customer
Access	R
Value	dwDeviation

2.4.3 DCM Bus Shift Object 0x2102

Object Type: RECORD, Manufacturer Specific Identity 0x55

Subindex	Description	Data Type	Access	M/O/C	Available
0	Number of Entries	Unsigned8	R	C	X
1	Error Code	Unsigned32	R	C	X
2	DC synchronized	Unsigned32	R	C	X
3	DCM controller synchronized	Unsigned32	R	C	X
4	Controller Set Value [nsec]	Signed32	R	C	X
5	Controller Error Filtered [nsec]	Signed32	R	C	X
6	Controller Error Average [nsec]	Signed32	R	C	X
7	Controller Error Maximum [nsec]	Signed32	R	C	X

Datatype for application access:

```
typedef struct _EC_T_OBJ2102 {
    EC_T_WORD         wSubIndex0;
    EC_T_DWORD        dwErrorCode;
    EC_T_BOOL         bDcInSync;
    EC_T_BOOL         bDcmInSync;
    EC_T_INT          nCtlSetVal;
    EC_T_INT          nCtlErrorFilt;
    EC_T_INT          nCtlErrorAvg;
    EC_T_INT          nCtlErrorMax;
} EC_PACKED(1) EC_T_OBJ2102;
```

2.4.4 Bus Load Base Object 0x2200

Object Type: RECORD, Manufacturer Specific Identity 0x44

Subindex	Description	Data Type	Access	M/O/C	Available
0	Number of Entries	Unsigned8	R	C	X
1	TX bytes/second actual value	Unsigned32	R	C	X
2	TX bytes/second min. value	Unsigned32	R	C	X
3	TX bytes/second max. value	Unsigned32	R	C	X
4	TX bytes/cycle actual value	UnSigned32	R	C	X
5	TX bytes/cycle min. value	UnSigned32	R	C	X
6	TX bytes/cycle max. value	UnSigned32	R	C	X
7	Reset Bus load Counters	Unsigned16	RW	C	A

Datatype for application access:

```
typedef struct _EC_T_OBJ2200 {
    EC_T_WORD         wSubIndex0;
    EC_T_DWORD        dwBytesPerSecondAct;
    EC_T_DWORD        dwBytesPerSecondMin;
    EC_T_DWORD        dwBytesPerSecondMax;
    EC_T_DWORD        dwBytesPerCycleAct;
    EC_T_DWORD        dwBytesPerCycleMin;
    EC_T_DWORD        dwBytesPerCycleMax;

    EC_T_WORD         wClearCounters;
} EC_PACKED(1) EC_T_OBJ2200;
```

wClearCounters is used to clear the counters of the object.

Bit 0..6: Clear corresponding Counter ID

- Bit 0: Clear all Counters
- Bit 1: Clear dwBytesPerSecondAct
- Bit 2: Clear dwBytesPerSecondMin
- Bit 3: Clear dwBytesPerSecondMax
- Bit 4: Clear dwBytesPerCycleAct
- Bit 5: Clear dwBytesPerCycleMin
- Bit 6: Clear dwBytesPerCycleMax

2.5 Slave specific objects

The Slave specific object structures are repeated for each slave. The number of slave specific objects may vary with different EC-Master versions. For version 1.5.0 and up the limit are 512 slave info objects.

To access a specific slave entry an application evaluates the correct object index by adding the Slave ID to the Corresponding base address (i.E. 0xA001 for diagnosis data of the second slave).

Note for entry 0x3000 (Slave information objects): The order of the slave in this object depends on the order of the slaves on the bus. The slave entry must therefore not be accessed by 0x3000 + <slave id>, instead the position within the EtherCAT bus is needed.

The following example shows how to determine the entry index of the corresponding slave entry:

<entry index> = 0x3000 + (<slave auto increment address> * -1)

Vendor specific Slave Configuration / Information Objects: 0x3000-0x3FFF

0x3000 - 0x3FFF	RECORD	One index entry for each configured slave (from ENI)	SlaveCfgInfo (0x43)	C	X

Slave Configuration Objects: 0x8000-0x8FFF

0x8000 - 0x8FFF	RECORD	One index entry for each configured slave (from ENI)	SlaveCfg (0x45)	M	X

Slave Information Objects: 0x9000-0x9FFF

0x9000 - 0x9FFF	RECORD	One index entry for each connected slave (updated during bus scan)	SlaveInfo (0x46)	O	X

Slave Diagnosis Data Objects: 0xA000-0xAFFF

0xA000 - 0xAFFF	RECORD	One subindex entry for each connected slave (cyclic updated)	SlaveDiag (0x47)	O	X

2.5.1 *Slave Configuration and Information Object 0x3000-0x3FFF*

Object Type: RECORD, Manufacturer Specific Identity 0x43

Subindex	Description	Data Type	Access	M/O/C	Available
0	Number of Entries	Unsigned8	R	M	X
1	Entry Valid	Bool32	R	M	X
2	VendorId (Bus)	Unsigned32	R	M	X
3	ProductCode (Bus)	Unsigned32	R	M	X
4	Revision No (Bus)	Unsigned32	R	M	X

5	Serial No (Bus)	Unsigned32	R	M	X
6	Device Name (Config)	Visible_String[80]	R	M	X
7	Auto Increment Address (Bus)	Unsigned16	R	M	X
8	Station Address ESC register: 0x0010	Unsigned16	R	M	X
9	Configured Station Address ENI file value “PhysAddr”	Unsigned16	R	M	X
10	Station Alias Address ESC register: 0x0012	Unsigned16	R	M	X
11	PortState (Bus)	Unsigned16	R	M	X
12	DC Support (Bus)	Bool32	R	M	X
13	DC Support 64Bit (Bus)	Bool32	R	M	X
14	Mailbox Support (Config)	Bool32	R	M	X
15	Requested State (Slave instance)	Unsigned32	RW	M	X
16	Current State (Slave instance)	Unsigned32	R	M	X
17	Error Flag Set (Slave instance)	Bool32	R	M	X
18	Enable Linkmessages (Slave instance)	Bool32	RW	M	X
19	Error code (slave instance)	Unsigned32	R	M	X
20	Sync Pulse active (Config, slave instance)	Bool32	R	M	X
21	DC Sync 0 Period (Config, slave instance)	Unsigned32	R	M	X
22	DC Sync 1 Period (Config, slave instance)	Unsigned32	R	O	X
23	SB Error Code (Bus Topology)	Unsigned32	R	O	X
24	RX Error Counter Port 0 (Bus)	Unsigned16	R	M	X
25	RX Error Counter Port 1 (Bus)	Unsigned16	R	M	X
26	RX Error Counter Port 2 (Bus)	Unsigned16	R	M	X
27	RX Error Counter Port 3 (Bus)	Unsigned16	R	M	X
28	Forwarded RX Error Counter Port 0 (Bus)	Unsigned8	R	M	X
29	Forwarded RX Error Counter Port 1 (Bus)	Unsigned8	R	M	X
30	Forwarded RX Error Counter Port 2 (Bus)	Unsigned8	R	M	X
31	Forwarded RX Error Counter Port 3 (Bus)	Unsigned8	R	M	X
32	EtherCAT Processing Unit Error Counter (Bus)	Unsigned8	R	M	X
33	PDI Errror Counter (Bus)	Unsigned8	R	M	X

34	Support Mailbox Protocols Bit 0=AoE, Bit 1=EoE, Bit 2=CoE, Bit 3=FoE, Bit 4=SoE, Bit 5=VoE	Unsigned16	R	O	X
35	Lost Link Counter Port 0 (Bus)	Unsigned8	R	M	X
36	Lost Link Counter Port 1 (Bus)	Unsigned8	R	M	X
37	Lost Link Counter Port 2 (Bus)	Unsigned8	R	M	X
38	Lost Link Counter Port 3 (Bus)	Unsigned8	R	M	X
39	FMMU’s supported (Bus)	Unsigned8	R	M	X
40	Sync Managers supported (Bus)	Unsigned8	R	M	X
41	RAM Size in kByte (Bus)	Unsigned8	R	M	X
42	Port Descriptor (Bus)	Unsigned8	R	M	X
43	ECS Type (Config)	Unsigned8	R	M	X
44	Slave is optional (Config)	Bool32	R	M	X
45	Slave is present (Bus)	Bool32	R	M	X
46	Hot connect group ID	Unsigned32	R	M	X
47	System time difference	Unsigned32	R	M	X
48	Process Data Input Offset [Bit]	Unsigned32	R	M	X
49	Process Data Input Size [Bit]	Unsigned32	R	M	X
50	Process Data Output Offset [Bit]	Unsigned32	R	M	X
51	Process Data Output Size [Bit]	Unsigned32	R	M	X

Datatype for application access:

```
typedef struct _EC_T_OBJ3XXX {
        EC_T_WORD               wSubIndex0;
        EC_T_BOOL               bEntryValid;
        EC_T_DWORD              dwVendorID;
        EC_T_DWORD              dwProductCode;
        EC_T_DWORD              dwRevisionNo;
        EC_T_DWORD              dwSerialNo;
        EC_T_CHAR               szDeviceName[MAX_SLAVE_DEVICENAME];
        EC_T_WORD               wAutoIncAddr;
        EC_T_WORD               wPhysAddr;
        EC_T_WORD               wConfigPhysAddr;
        EC_T_WORD               wAliasAddr;
        EC_T_WORD               wPortState;
        EC_T_BOOL               bDCSupport;
        EC_T_BOOL               bDC64Support;
        EC_T_BOOL               bMailboxSupport;
        EC_T_DWORD              dwState;
        EC_T_DWORD              dwReqState;
        EC_T_BOOL               bErrFlagSet;
        EC_T_BOOL               bEnableLinkMsgs;
        EC_T_DWORD              dwErrorCode;
        EC_T_BOOL               bSyncPulseActive;
        EC_T_DWORD              dwDCSync0Period;
        EC_T_DWORD              dwDCSync1Period;
        EC_T_DWORD              dwSBErrorCode;
        EC_T_WORD               wRxErrorCounter0;
        EC_T_WORD               wRxErrorCounter1;
        EC_T_WORD               wRxErrorCounter2;
        EC_T_WORD               wRxErrorCounter3;
        EC_T_BYTE               byFwdRxErrorCounter0;
        EC_T_BYTE               byFwdRxErrorCounter1;
        EC_T_BYTE               byFwdRxErrorCounter2;
        EC_T_BYTE               byFwdRxErrorCounter3;
        EC_T_BYTE               byEcatProcUnitErrorCounter;
        EC_T_BYTE               byPDIErrorCounter;
        EC_T_WORD               wMbxSupportedProtocols;
        EC_T_BYTE               byLostLinkCounter0;
        EC_T_BYTE               byLostLinkCounter1;
        EC_T_BYTE               byLostLinkCounter2;
        EC_T_BYTE               byLostLinkCounter3;
        EC_T_BYTE               byFmmusSupported;
        EC_T_BYTE               bySyncManagersSupported;
        EC_T_BYTE               byRamSizeKb;
        EC_T_BYTE               byPortDescriptor;
        EC_T_BYTE               byESCType;
        EC_T_BOOL               bSlaveIsOptional;
        EC_T_BOOL               bSlaveIsPresent;
        EC_T_DWORD              dwHotConnectGroupId;
        EC_T_DWORD              dwSystemTimeDifference;
        EC_T_DWORD              dwPdOffsIn;
        EC_T_DWORD              dwPdSizeIn;
        EC_T_DWORD              dwPdOffsOut;
        EC_T_DWORD              dwPdSizeOut;
} EC_PACKED(1) EC_T_OBJ3XXX;
```

2.5.2 Slave Configuration Objects: 0x8000-0x8FFF

Object Type: RECORD, Manufacturer Specific Identity 0x45

The Configuration Data at 0x8nnn contains the information about the configured EtherCAT slaves.

Subindex	Description	Data Type	Access	M/O/C	Available
0	Number of Entries	Unsigned8	R	M	X
1	Configured Station Address ENI file value “PhysAddr”	Unsigned16	R	M	X
2	Type	Visible_String[64]	RW	M	X
3	Name	Visible_String[64]	R	M	X
4	Device Type	Unsigned32	R	M	X
5	Vendor ID	Unsigned32	R	M	X
6	Product Code	Unsigned32	R	M	X
7	Revision Number	Unsigned32	R	M	X
8	Version Number	Unsigned32	R	M	X
33	Mailbox Out Size (if mailbox slave)	Unsigned16	R	M	X
34	Mailbox In Size (if mailbox slave)	Unsigned16	R	M	X
35	Link Status	Unsigned8	R	O	-
36	Link Present	Unsigned8	R	M	X
37	Flags	Unsigned8	R	M	X

Datatype for application access:

```
typedef struct _EC_T_OBJ8XXX {
    EC_T_WORD               wSubIndex0;
    EC_T_WORD               wFixedStationAddr;
    EC_T_CHAR               szType[64];
    EC_T_CHAR               szName[64];
    EC_T_DWORD              dwDeviceType;
    EC_T_DWORD              dwVendorID;
    EC_T_DWORD              dwProductCode;
    EC_T_DWORD              dwRevision;
    EC_T_DWORD              dwSerial;
    EC_T_WORD               wMailboxOutSize;
    EC_T_WORD               wMailboxInSize;
    EC_T_BYTE               byLinkPreset;
    EC_T_BYTE               byFlags;
} EC_PACKED(1) EC_T_OBJ8XXX;
```

2.5.3 Slave Information Objects: 0x9000-0x9FFF

Object Type: RECORD, Manufacturer Specific Identity 0x46

The information data at 0x9nnn contains the information about the connected EtherCAT slaves. Executing the Scan Bus command (0xF002) will refresh them.

Subindex	Description	Data Type	Access	M/O/C	Available
0	**Number of Entries**	Unsigned8	R	M	X
1	**Fixed Station Address** of the Nth EtherCAT Slave found (same value as 0xF040:0n)	Unsigned16	R	M	X
5	**Vendor ID** of the Nth EtherCAT Slave found (entry 0x1018:01 of the EtherCAT slave)	Unsigned32	R	M	X
6	**Product Code** of the Nth EtherCAT Slave found (entry 0x1018:02 of the EtherCAT slave)	Unsigned32	R	M	X
7	**Revision Number** of the first EtherCAT Slave found (entry 0x1018:03 of the EtherCAT slave)	Unsigned32	R	M	X
8	**Serial Number** of the first EtherCAT Slave found (entry 0x1018:04 of the EtherCAT slave)	Unsigned32	R	M	X
32	**DL Status** (Register 0x110-0x111) of the Nth EtherCAT slave found.	Unsigned16	R	M	X

Datatype for application access:

```
typedef struct _EC_T_OBJ9XXX {
    EC_T_WORD               wSubIndex0;
    EC_T_WORD               wFixedStationAddr;
    EC_T_DWORD              dwVendorID;
    EC_T_DWORD              dwProductCode;
    EC_T_DWORD              dwRevision;
    EC_T_DWORD              dwSerial;
    EC_T_WORD               wDLStatus;
} EC_PACKED(1) EC_T_OBJ9XXX;
```

2.5.4 Slave Diagnosis Data Objects: 0xA000-0xAFFF

Object Type: RECORD, Manufacturer Specific Identity 0x47

The diagnosis data at 0xAnnn contains enhanced information about the connected EtherCAT slaves including topology information.

Subindex	Description	Data Type	Access	M/O/C	Available
0	**Number of Entries**	Unsigned8	R	M	X
1	**AL Status** (Register 0x130-0x131) of the Nth EtherCAT slave configured.	Unsigned16	R	M	X
2	**AL Control** (Register 0x120-0x121) of the Nth EtherCAT slave configured.	Unsigned16	RW	M	X
3	**Last AL Status Code** (Register 0x0134-0x0135) of the Nth EtherCAT slave configured.	Unsigned16	R	M	X
4	**Link Conn. Status** Actual link status of the Nth EtherCAT slave configured.	Unsigned8	R	O	X
5	**Link Control** (Register 0x0101) of the Nth EtherCAT slave configured.	Unsigned8	RW	O	X
6	**Fixed Address Conn. Port 0** (Register 0x0010-0x0011) of the slave connected to port 0 of the Nth EtherCAT slave configured.	Unsigned16	R	M	X
7	**Fixed Address Conn. Port 1**	Unsigned16	R	M	X
8	**Fixed Address Conn. Port 2**	Unsigned16	R	M	X
9	**Fixed Address Conn. Port 3**	Unsigned16	R	M	X
10	**CRC Error Counter Port 0** Number of CRC errors counted by port 0 of the Nth EtherCAT slave configured.	Unsigned32	R	M	X
11	**CRC Error Counter Port 1**	Unsigned32	R	M	X
12	**CRC Error Counter Port 2**	Unsigned32	R	M	X
13	**CRC Error Counter Port 3**	Unsigned32	R	M	X
14	**Cyclic WC Error Counter** Number of cycles the datagram for process data exchange of the Nth EtherCAT slave was received with an invalid Working Counter.	Unsigned32	R	M	X
15	**Slave Not Present Counter** Number of times the master lost physical link with the Nth EtherCAT slave.	Unsigned32	R	O	X
16	**Abnormal State Change Counter** Number of times the Nth EtherCAT slave left its state autonomously to a lower state without master request.	Unsigned32	R	M	X
17	**Disable Automatic Link Support**	Boolean	RW	O	X

Datatype for application access:

```
typedef struct _EC_T_OBJAXXX {
    EC_T_WORD               wSubIndex0;
    EC_T_WORD               wALStatus;
    EC_T_WORD               wALControl;
    EC_T_WORD               wLastALStatusCode;
    EC_T_BYTE               byLinkConnStatus;
    EC_T_BYTE               byLinkControl;
    EC_T_WORD               wFixedAddressConnPort0;
    EC_T_WORD               wFixedAddressConnPort1;
    EC_T_WORD               wFixedAddressConnPort2;
    EC_T_WORD               wFixedAddressConnPort3;
    EC_T_DWORD              dwCRCErrorCounterPort0;
    EC_T_DWORD              dwCRCErrorCounterPort1;
    EC_T_DWORD              dwCRCErrorCounterPort2;
    EC_T_DWORD              dwCRCErrorCounterPort3;
    EC_T_DWORD              dwCyclicWCErrorCounter;
    EC_T_DWORD              dwSlaveNotPresentCounter;
    EC_T_DWORD              dwAbnormalStateCounter;
    EC_T_BYTE               bDisableAutomaticLinkControl;
} EC_PACKED(1) EC_T_OBJAXXX;
```

2.6 Device Area Objects: 0xF000-0xFFFF

Device Area Objects: 0xF000-0xFFFF

0xF000	RECORD	Modular Device Profile	DeviceProfile (0x48)	M	X
0xF002	RECORD	Detect Modules Command	DetectModCmd (0x49)	M	X
0xF020 - 0xF02F	RECORD	Configured Address List	ConfAddrList (0x50)	M	X
0xF040 - 0xF04F	RECORD	Detected Address List	ConnAddrList (0x51)	O	X
0xF120	RECORD	Frame Statistics	FrameStatistics (0x56)	M	X
0xF200	RECORD	Diag Interface Control	DiagInterfaceCtl (0x57)	M	X

2.6.1 Modular Device Profile Object 0xF000

Object Type: RECORD, Manufacturer Specific Identity 0x48

Subindex	Description	Data Type	Access	M/O/C	Available
0	**Number of Entries**	Unsigned8	R	M	X
1	**Index distance** between two modules. This value is always read as 1.	Unsigned16	R	M	X
2	**Maximum number of EtherCAT slaves** connected to the EtherCAT bus. This value is read as 512.	Unsigned16	R	M	X
3	**Available entries in objects 0x8xxx** (number of configured slaves).	Unsigned32	R	M	X
4	**Available entries in objects 0x9xxx** (number of connected slaves).	Unsigned32	R	M	X

Datatype for application access:

```
typedef struct _EC_T_OBJF000_PRIV {
    EC_T_WORD               wSubIndex0;
    EC_T_WORD               wIndexDistance;
    EC_T_WORD               wMaxModuleCnt;
    EC_T_DWORD              dwGeneralCfg;
    EC_T_DWORD              dwGeneralInfo;
} EC_PACKED(1) EC_T_OBJF000_PRIV;
```

2.6.2 Detect Modules Command Object 0xF002

Object Type: RECORD, Manufacturer Specific Identity 0x49

Subindex	Description	Data Type	Access	M/O/C	Available
0	**Number of Entries**	Unsigned8	R	M	X
1	**Scan Command Request** When this subindex is written, the EtherCAT Master shall scan the EtherCAT bus and update the objects 0xF04x and 0x9nnn	Unsigned16	RW	M	X
2	**Scan Command Status** 1: command is finished, no error 3: command is finished, error 100-199: 0-99% of the command is done 255: command is executing	Unsigned8	R	M	X
3	**Scan Command Response** Byte 0:Scan Command Status Byte 1: always 0 Byte 2-3: 0: no error, > 0: vendor specific error code Byte 4-5: number of EtherCAT slaves found	Unsigned48	R	M	X

Datatype for application access:

```
typedef struct _EC_T_OBJF002 {
    EC_T_WORD               wSubIndex0;
    EC_T_BYTE               abyCmdRequest[2];
    EC_T_WORD               byCmdStatus;
    EC_T_BYTE               abyCmdResponse[6];
} EC_PACKED(1) EC_T_OBJF002;
```

2.6.3 Configured Address List Object 0xF020-0xF02F

Object Type: RECORD, Manufacturer Specific Identity 0x50

The object range 0xF020-0xF02F contains objects with each at maximum 128 entries for all configured slaves.

The last 0xF02n object might be shorter than the first one.

Index / Subindex	Description	Data Type	Access	M/O/C	Available
0xF020 / 0	**Number of Entries**	**Unsigned8**	**R**	**M**	**X**
0xF020 / 1	Fixed Station Address of the first EtherCAT Slave configured.	Unsigned16	R	M	X
0xF020 / 2	Fixed Station Address of the second EtherCAT Slave configured.	Unsigned16	R	M	X
…	…	Unsigned16	R	M	X
0xF020 / 128	Fixed Station Address of the 128th EtherCAT Slave configured.	Unsigned16	R	M	X
0xF021 / 0	**Number of Entries**	**Unsigned8**	**R**	**M**	**X**
0xF021 / 1	Fixed Station Address of the 129th EtherCAT Slave configured.	Unsigned16	R	M	X
…	…				

2.6.4 Detected Address List Object 0xF040-0xF04F

Object Type: RECORD, Manufacturer Specific Identity 0x51

The object range 0xF040-0xF04F contains objects with each at maximum 128 entries according to the max. bus slave count (see emInitMaster()).

The last 0xF04n object might be shorter than the first one.

Index / Subindex	Description	Data Type	Access	M/O/C	Available
0xF040 / 0	**Number of Entries**	**Unsigned8**	**R**	**M**	**X**
0xF040 / 1	Fixed Station Address of the first EtherCAT Slave detected.	Unsigned16	R	M	X
0xF040 / 2	Fixed Station Address of the second EtherCAT Slave detected.	Unsigned16	R	M	X
…	…	Unsigned16	R	M	X
0xF040 / 128	Fixed Station Address of the 128th EtherCAT Slave detected.	Unsigned16	R	M	X
0xF041 / 0	**Number of Entries**	**Unsigned8**	**R**	**M**	**X**

0xF041 / 1	Fixed Station Address of the 129th EtherCAT Slave detected.	Unsigned16	R	M	X
…	…				

2.6.5 Frame Statistics Object 0xF120

Object Type: RECORD, Manufacturer Specific Identity 0x56

Subindex	Description	Data Type	Access	M/O/C	Available
0	**Number of Entries**	Unsigned8	R	M	X
1	**Cyclic Lost Frames** Number of cyclic lost frames.	Unsigned32	R	M	X
2	**Acyclic Lost Frames** Number of acyclic lost frames.	Unsigned32	R	M	X

Datatype for application access:

```
typedef struct _EC_T_OBJF120 {
    EC_T_WORD               wSubIndex0;
    EC_T_DWORD              dwCyclicLostFrames;
    EC_T_DWORD              dwAcyclicLostFrames;
} EC_PACKED(1) EC_T_OBJF120;
```

2.6.6 Diag Interface Control Object 0xF200

Object Type: RECORD, Manufacturer Specific Identity 0x57

Subindex	Description	Data Type	Access	M/O/C	Available
0	**Number of Entries**	Unsigned8	R	M	X
16	**Reset Diag Info** resets values to 0 of objects 0xAnnn SI 03, 0xAnnn SI 04, 0xAnnn SI05-08, 0xAnnn SI08, 0xAnnn SI09, 0xF120 SI01 and 0xF120 SI01	Boolean	RW	M	X

Datatype for application access:

```
typedef struct _EC_T_OBJF200 {
    EC_T_WORD               wSubIndex0;
    EC_T_BOOL               bResetDiagInfo;
} EC_PACKED(1) EC_T_OBJF200;
```