

**Migration Guide**  
**EtherCAT Slave**  
**Migration from Version 2.5 to 4.2**

**Hilscher Gesellschaft für Systemautomation mbH**

**[www.hilscher.com](http://www.hilscher.com)**

DOC120612MG03EN | Revision 3 | English | 2013-10 | Preliminary | Public

# Table of Contents

<b>1</b>	<b>Introduction.....</b>	<b>4</b>
1.1	About this Document.....	4
1.2	List of Revisions .....	4
1.3	References .....	4
<b>2</b>	<b>Changes of V2.5.x Packets.....</b>	<b>5</b>
2.1	ESM Packets.....	5
2.1.1	Removed Packets .....	5
2.1.2	Replaced Packets .....	6
2.1.3	ECAT_ESM_SII_WRITE_REQ / ECAT_ESM_SII_WRITE_CNF .....	7
2.1.4	ECAT_ESM_SII_READ_REQ / ECAT_ESM_SII_READ_CNF .....	7
2.1.5	ECAT_ESM_SII_UPDATE_VENDOR_DATA_IND / ECAT_ESM_SII_UPDATE_VENDOR_DATA_RES .....	7
2.1.6	ECAT_ESM_SII_WRITE_VENDOR_DATA_REQ / ECAT_ESM_SII_WRITE_VENDOR_DATA_CNF .....	8
2.1.7	ECAT_ESM_ALCONTROL_INIT_IND / ECAT_ESM_ALCONTROL_INIT_RES .....	8
2.1.8	ECAT_ESM_ALCONTROL_PRE_OPERATIONAL_IND / ECAT_ESM_ALCONTROL_PRE_OPERATIONAL_RES .....	8
2.1.9	ECAT_ESM_ALCONTROL_BOOTSTRAP_IND / ECAT_ESM_ALCONTROL_BOOTSTRAP_RES .....	8
2.1.10	ECAT_ESM_ALCONTROL_SAFE_OPERATIONAL_IND / ECAT_ESM_ALCONTROL_SAFE_OPERATIONAL_RES .....	8
2.1.11	ECAT_ESM_ALCONTROL_OPERATIONAL_IND / ECAT_ESM_ALCONTROL_OPERATIONAL_RES .....	8
2.1.12	ECAT_ESM_ALSTATUS_INIT_IND / ECAT_ESM_ALSTATUS_INIT_RES .....	8
2.1.13	ECAT_ESM_ALSTATUS_PRE_OPERATIONAL_IND / ECAT_ESM_ALSTATUS_PRE_OPERATIONAL_RES .....	9
2.1.14	ECAT_ESM_ALSTATUS_BOOTSTRAP_IND / ECAT_ESM_ALSTATUS_BOOTSTRAP_RES .....	9
2.1.15	ECAT_ESM_ALSTATUS_SAFE_OPERATIONAL_IND / ECAT_ESM_ALSTATUS_SAFE_OPERATIONAL_RES .....	9
2.1.16	ECAT_ESM_ALSTATUS_OPERATIONAL_IND / ECAT_ESM_ALSTATUS_OPERATIONAL_RES .....	9
2.1.17	ECAT_ESM_SETINIT_IND / ECAT_ESM_SETINIT_RES .....	9
2.1.18	ECAT_ESM_REGISTERNOTIFY_REQ / ECAT_ESM_REGISTERNOTIFY_CNF .....	9
2.1.19	ECAT_ESM_UNREGISTERNOTIFY_REQ / ECAT_ESM_UNREGISTERNOTIFY_CNF .....	10
2.1.20	ECAT_ESM_START_PROCDATA_INPUT_IND / ECAT_ESM_START_PROCDATA_INPUT_RES .....	10
2.1.21	ECAT_ESM_STOP_PROCDATA_INPUT_IND / ECAT_ESM_STOP_PROCDATA_INPUT_RES .....	10
2.1.22	ECAT_ESM_START_PROCDATA_OUTPUT_IND / ECAT_ESM_START_PROCDATA_OUTPUT_RES .....	10
2.1.23	ECAT_ESM_STOP_PROCDATA_OUTPUT_IND / ECAT_ESM_STOP_PROCDATA_OUTPUT_RES .....	10
2.1.24	ECAT_ESM_INIT_COMPLETE_IND / ECAT_ESM_INIT_COMPLETE_RES .....	10
2.1.25	ECAT_ESM_REGISTER_PROCDATA_INDICATIONS_REQ / ECAT_ESM_REGISTER_PROCDATA_INDICATIONS_CNF .....	10
2.1.26	ECAT_ESM_UNREGISTER_PROCDATA_INDICATIONS_REQ / ECAT_ESM_UNREGISTER_PROCDATA_INDICATIONS_CNF .....	10
2.1.27	ECAT_ESM_SET_QUEUE_CNF_AL_CONTROL_REQ / ECAT_ESM_SET_QUEUE_CNF_AL_CONTROL_CNF .....	11
2.1.28	ECAT_ESM_CLR_QUEUE_CNF_AL_CONTROL_REQ / ECAT_ESM_CLR_QUEUE_CNF_AL_CONTROL_CNF .....	11
2.1.29	ECAT_ESM_ALCONTROL_CHANGE_IND / ECAT_ESM_ALCONTROL_CHANGE_RES .....	11
2.1.30	ECAT_ESM_ALSTATUS_CHANGE_REQ / ECAT_ESM_ALSTATUS_CHANGE_CNF .....	11
2.1.31	ECAT_ESM_SET_AL_STATUS_REQ / ECAT_ESM_SET_AL_STATUS_CNF .....	11
2.1.32	ECAT_ESM_SET_SAFEOP_CHECK_FN_REQ / ECAT_ESM_SET_SAFEOP_CHECK_FN_CNF .....	11
2.1.33	ECAT_ESM_SET_OP_CHECK_FN_REQ / ECAT_ESM_SET_OP_CHECK_FN_CNF .....	11
2.2	DPM Packets .....	12
2.2.1	Replaced Packets .....	12
2.2.2	ECAT_DPM_SET_IO_SIZE_REQ / ECAT_DPM_SET_IO_SIZE_CNF .....	12
2.2.3	ECAT_DPM_WARMSTART_REQ / ECAT_DPM_WARMSTART_CNF .....	12
2.2.4	ECAT_DPM_SET_STATION_ALIAS_REQ / ECAT_DPM_SET_STATION_ALIAS_CNF .....	12
2.2.5	ECAT_DPM_GET_STATION_ALIAS_REQ / ECAT_DPM_GET_STATION_ALIAS_CNF .....	12
2.2.6	ECAT_DPM_SET_CONFIG_REQ / ECAT_DPM_SET_CONFIG_CNF .....	12
2.2.7	ECAT_DPM_SET_CONFIGURATION_REQ / ECAT_DPM_SET_CONFIGURATION_CNF .....	13
2.2.8	ECAT_DPM_SET_UPDATE_CFG_REQ / ECAT_DPM_SET_UPDATE_CFG_CNF .....	13
2.3	CoE Packets .....	14
2.3.1	Removed Packets .....	14
2.3.2	Replaced Packets .....	14

2.3.3	ECAT_COE_SEND_EMERGENCY_REQ / ECAT_COE_SEND_EMERGENCY_CNF .....	14
2.3.4	ECAT_LOCAL_SDO_DOWNLOAD_EXP_REQ / ECAT_LOCAL_SDO_DOWNLOAD_EXP_CNF .....	14
2.3.5	ECAT_LOCAL_SDO_UPLOAD_EXP_REQ / ECAT_LOCAL_SDO_UPLOAD_EXP_CNF .....	15
2.3.6	ECAT_SDO_DOWNLOAD_EXP_REQ / ECAT_SDO_DOWNLOAD_EXP_CNF .....	15
2.3.7	ECAT_SDO_UPLOAD_EXP_REQ / ECAT_SDO_UPLOAD_EXP_CNF .....	15
2.3.8	ECAT_COE_SDO_SET_TIMEOUTS_REQ / ECAT_COE_SDO_SET_TIMEOUTS_CNF .....	15
2.3.9	ECAT_COE_SDO_GET_TIMEOUTS_REQ / ECAT_COE_SDO_GET_TIMEOUTS_CNF .....	15
2.4	OD Packets .....	16
2.4.1	Removed Packets .....	16
2.4.2	Replaced Packets .....	16
2.4.3	ECAT_OD_SET_INDICATION_TIMEOUT_REQ / ECAT_OD_SET_INDICATION_TIMEOUT_CNF .....	17
2.4.4	ECAT_OD_GET_INDICATION_TIMEOUT_REQ / ECAT_OD_GET_INDICATION_TIMEOUT_CNF .....	17
2.4.5	ECAT_OD_CREATE_OBJECT_REQ / ECAT_OD_CREATE_OBJECT_CNF .....	17
2.4.6	ECAT_OD_CREATE_SUBOBJECT_REQ / ECAT_OD_CREATE_SUBOBJECT_CNF .....	19
2.4.7	ECAT_OD_DELETE_OBJECT_REQ / ECAT_OD_DELETE_OBJECT_CNF .....	20
2.4.8	ECAT_OD_CREATE_DATATYPE_REQ / ECAT_OD_CREATE_DATATYPE_CNF .....	20
2.4.9	ECAT_OD_DELETE_DATATYPE_REQ / ECAT_OD_DELETE_DATATYPE_CNF .....	20
2.4.10	ECAT_OD_NOTIFY_REGISTER_REQ / ECAT_OD_NOTIFY_REGISTER_CNF .....	21
2.4.11	ECAT_OD_NOTIFY_UNREGISTER_REQ / ECAT_OD_NOTIFY_UNREGISTER_CNF .....	21
2.4.12	ECAT_OD_NOTIFY_READ_IND / ECAT_OD_NOTIFY_READ_RES .....	21
2.4.13	ECAT_OD_NOTIFY_WRITE_IND / ECAT_OD_NOTIFY_WRITE_RES .....	21
2.4.14	ECAT_OD_UNDEFINED_NOTIFY_REGISTER_REQ / ECAT_OD_UNDEFINED_NOTIFY_REGISTER_CNF .....	21
2.4.15	ECAT_OD_UNDEFINED_NOTIFY_UNREGISTER_REQ / ECAT_OD_UNDEFINED_NOTIFY_UNREGISTER_CNF .....	22
2.4.16	ECAT_OD_UNDEFINED_READ_PREPARE_IND / ECAT_OD_UNDEFINED_READ_PREPARE_RES .....	22
2.4.17	ECAT_OD_UNDEFINED_READ_DATA_IND / ECAT_OD_UNDEFINED_READ_DATA_RES .....	22
2.4.18	ECAT_OD_UNDEFINED_WRITE_DATA_IND / ECAT_OD_UNDEFINED_WRITE_DATA_RES .....	22
2.4.19	ECAT_OD_SDOINFO_REGISTER_REQ / ECAT_OD_SDOINFO_REGISTER_CNF .....	22
2.4.20	ECAT_OD_SDOINFO_UNREGISTER_REQ / ECAT_OD_SDOINFO_UNREGISTER_CNF .....	22
2.4.21	ECAT_OD_SDOINFO_GET_LIST_IND / ECAT_OD_SDOINFO_GET_LIST_RES .....	23
2.4.22	ECAT_OD_SDOINFO_GET_OBJ_DESC_IND / ECAT_OD_SDOINFO_GET_OBJ_DESC_RES .....	24
2.4.23	ECAT_OD_SDOINFO_GET_ENTRY_DESC_IND / ECAT_OD_SDOINFO_GET_ENTRY_DESC_RES .....	24
2.4.24	ECAT_OD_NOTIFY_REGISTER_RESET_REQ / ECAT_OD_NOTIFY_REGISTER_RESET_CNF .....	25
2.4.25	ECAT_OD_SET_OBJECT_NAME_REQ / ECAT_OD_SET_OBJECT_NAME_CNF .....	25
2.4.26	ECAT_OD_SET_SUBOBJECT_NAME_REQ / ECAT_OD_SET_SUBOBJECT_NAME_CNF .....	25
3	<b>New Packets.....</b>	<b>26</b>
3.1	General Packets.....	26
3.1.1	ECAT_SET_CONFIG_REQ / ECAT_SET_CONFIG_CNF .....	26
3.2	ESM Packets.....	26
3.2.1	ECAT_ESM_REGISTER_FOR_SIIWRITE_INDICATIONS_REQ / ECAT_ESM_REGISTER_FOR_SIIWRITE_INDICATIONS_CNF .....	26
3.2.2	ECAT_ESM_UNREGISTER_FROM_SIIWRITE_INDICATIONS_REQ / ECAT_ESM_UNREGISTER_FROM_SIIWRITE_INDICATIONS_CNF .....	26
3.2.3	ECAT_ESM_SII_WRITE_IND / ECAT_ESM_SII_WRITE_RES .....	26
3.2.4	ECAT_ESM_ALSTATUS_CHANGED_IND / ECAT_ESM_ALSTATUS_CHANGED_RES .....	26
3.2.5	ECAT_ESM_GET_ALSTATUS_REQ / ECAT_ESM_GET_ALSTATUS_CNF .....	26
4	<b>Changes in Error Codes .....</b>	<b>27</b>
5	<b>Appendix .....</b>	<b>28</b>
5.1	List of Tables .....	28
5.2	Contacts .....	29

# 1 Introduction

## 1.1 About this Document

This migration guide gives a brief overview on the changes between the user interfaces of the EtherCAT Slave Stack V2.5.x and the new EtherCAT Slave Stack V4.2.x with the aim to support and lead you through the integration process of the given stack into your own application. The guide covers the following topics:

- Changes of V2.5.x Packets
- New Packets
- Changes in Error Codes

## 1.2 List of Revisions

Rev	Date	Name	Chapter	Revision
1	2012-06-18	RW/RG	All	Created
2	2012-07-03	RW/RG/ SB		A lot of causes of removed and replacements for replaced packets have been added.
3	2013-10-01	RW/RG		Some clarifications and corrections

Table 1: List of Revisions

## 1.3 References

This document is based on the following manuals:

Hilscher Gesellschaft für Systemautomation mbH: EtherCAT Slave V4 Protocol API Manual, Revision 3, English, 2013

Hilscher Gesellschaft für Systemautomation mbH: Object Dictionary V3 Protocol API Manual, Revision 2, English, 2012.

Table 2: References

## 2 Changes of V2.5.x Packets

This chapter will give a short overview about the services of the stack V2.5.x and how the services changed compared to the stack V4.2.x. Even if the service is marked as "did not change compared to V2.5.x" it may happen that the names of an element inside the packet changed. As reference for comparison EtherCAT Slave Stack V2.5.16.0 is used.

### 2.1 ESM Packets

#### 2.1.1 Removed Packets

The following ESM packets of V2.5 have been removed as the respective service does no longer exist in V4.2.x:

Command Code	Packet Name	Cause of Removal
0x1984/ 0x1985	ECAT_ESM_START_PROCDATA_INPUT_IND / ECAT_ESM_START_PROCDATA_INPUT_RES	1
0x1986/ 0x1987	ECAT_ESM_STOP_PROCDATA_INPUT_IND / ECAT_ESM_STOP_PROCDATA_INPUT_RES	1
0x1988/ 0x1989	ECAT_ESM_START_PROCDATA_OUTPUT_IND / ECAT_ESM_START_PROCDATA_OUTPUT_RES	1
0x198A/ 0x198B	ECAT_ESM_STOP_PROCDATA_OUTPUT_IND / ECAT_ESM_STOP_PROCDATA_OUTPUT_RES	1
0x1990/ 0x1991	ECAT_ESM_REGISTER_PROCDATA_INDICATIONS_REQ / ECAT_ESM_REGISTER_PROCDATA_INDICATIONS_CNF	2
0x1992/ 0x1993	ECAT_ESM_UNREGISTER_PROCDATA_INDICATIONS_REQ / ECAT_ESM_UNREGISTER_PROCDATA_INDICATIONS_CNF	2
0x1BFC/ 0x1BFD	ECAT_ESM_SET_SAFEOP_CHECK_FN_REQ / ECAT_ESM_SET_SAFEOP_CHECK_FN_CNF	3
0x1BFE/ 0x1BFF	ECAT_ESM_SET_OP_CHECK_FN_REQ / ECAT_ESM_SET_OP_CHECK_FN_CNF	3

Table 3: Removed ESM Packets V2.5 => V4

The following table informs about the causes of the removal of a specific packet:

Cause #	Description of Cause
1	Used for packet API only! no need for application to enable/disable sync managers on this indication Sync managers are enabled/disabled inside V4.2.x stack (for the default case that sync manager 2 is used for EtherCAT master outputs and sync manager 3 is used for EtherCAT master inputs).
2	Used for packet API only! not needed any more because indication packets for start/stop of process data update moved from public interface to ESM task (sync managers 2 and 3 are enabled/disabled inside stack and not by user application)
3	Used for packet API only! replaced by internal sync manager configuration checks If functions were used for more than just sync manager configuration checks (eg. additional applications checks). Additional applications checks (eg. DC configuration checks) have to be moved to indication handling of ECAT_ESM_ALCONTROL_CHANGED_IND.

Table 4: Causes of Removal of ESM Packets V2.5 => V4

## 2.1.2 Replaced Packets

The following ESM packets of V2.5.x have been replaced in V4.2.x:

Command Code	Packet Name
0x1916/0x1917=> 0x1B80/0x1B81	ECAT_ESM_SII_UPDATE_VENDOR_DATA_IND / ECAT_ESM_SII_UPDATE_VENDOR_DATA_RES=> ECAT_ESM_SII_WRITE_IND / ECAT_ESM_SII_WRITE_RES (See 2.1.5)
0x1918/0x1919 => 0x1912/0x1913	ECAT_ESM_SII_WRITE_VENDOR_DATA_REQ / ECAT_ESM_SII_WRITE_VENDOR_DATA_CNF => ECAT_ESM_SII_WRITE_REQ / ECAT_ESM_SII_WRITE_CNF (See 2.1.6)
0x1962/0x1963 => 0x1B1C/0x1B1D	ECAT_ESM_ALCONTROL_INIT_IND / ECAT_ESM_ALCONTROL_INIT_RES=> ECAT_ESM_ALCONTROL_CHANGED_IND/ ECAT_ESM_ALCONTROL_CHANGED_RES (See 2.1.7)
0x1964/0x1965 => 0x1B1C/0x1B1D	ECAT_ESM_ALCONTROL_PRE_OPERATIONAL_IND / ECAT_ESM_ALCONTROL_PRE_OPERATIONAL_RES => ECAT_ESM_ALCONTROL_CHANGED_IND/ ECAT_ESM_ALCONTROL_CHANGED_RES (See 2.1.8)
0x1966/0x1967 => 0x1B1C/0x1B1D	ECAT_ESM_ALCONTROL_BOOTSTRAP_IND / ECAT_ESM_ALCONTROL_BOOTSTRAP_RES => ECAT_ESM_ALCONTROL_CHANGED_IND/ ECAT_ESM_ALCONTROL_CHANGED_RES (See 2.1.10)
0x1968/0x1969 => 0x1B1C/0x1B1D	ECAT_ESM_ALCONTROL_SAFE_OPERATIONAL_IND / ECAT_ESM_ALCONTROL_SAFE_OPERATIONAL_RES => ECAT_ESM_ALCONTROL_CHANGED_IND/ ECAT_ESM_ALCONTROL_CHANGED_RES (See 2.1.10)
0x1970/0x1971 => 0x1B1C/0x1B1D	ECAT_ESM_ALCONTROL_OPERATIONAL_IND / ECAT_ESM_ALCONTROL_OPERATIONAL_RES => ECAT_ESM_ALCONTROL_CHANGED_IND/ ECAT_ESM_ALCONTROL_CHANGED_RES (See 2.1.11)
0x1962/0x1963 => 0x19DE/0x19DF	ECAT_ESM_ALSTATUS_INIT_IND / ECAT_ESM_ALSTATUS_INIT_RES ECAT_ESM_ALSTATUS_CHANGED_IND / ECAT_ESM_ALSTATUS_CHANGED_RES See 2.1.12
0x1964/0x1965 => 0x19DE/0x19DF	ECAT_ESM_ALSTATUS_PRE_OPERATIONAL_IND / ECAT_ESM_ALSTATUS_PRE_OPERATIONAL_RES ECAT_ESM_ALSTATUS_CHANGED_IND / ECAT_ESM_ALSTATUS_CHANGED_RES See 2.1.13
0x1966/0x1967 => 0x19DE/0x19DF	ECAT_ESM_ALSTATUS_BOOTSTRAP_IND / ECAT_ESM_ALSTATUS_BOOTSTRAP_RES ECAT_ESM_ALSTATUS_CHANGED_IND / ECAT_ESM_ALSTATUS_CHANGED_RES See 2.1.13
0x1968/0x1969 => 0x19DE/0x19DF	ECAT_ESM_ALSTATUS_SAFE_OPERATIONAL_IND / ECAT_ESM_ALSTATUS_SAFE_OPERATIONAL_RES ECAT_ESM_ALSTATUS_CHANGED_IND / ECAT_ESM_ALSTATUS_CHANGED_RES See 2.1.15
0x1970/0x1971 => 0x19DE/0x19DF	ECAT_ESM_ALSTATUS_OPERATIONAL_IND / ECAT_ESM_ALSTATUS_OPERATIONAL_RES=> ECAT_ESM_ALSTATUS_CHANGED_IND / ECAT_ESM_ALSTATUS_CHANGED_RES See 2.1.16
0x1980/0x1981 => 0x19DE/0x19DF	ECAT_ESM_SETINIT_IND / ECAT_ESM_SETINIT_RES=> ECAT_ESM_SETREADY_REQ / ECAT_ESM_SETREADY_CNF Did not change in functionality See 2.1.17
0x1982/0x1983 => 0x2F10/0x2F11	ECAT_ESM_REGISTERNOTIFY_REQ / ECAT_ESM_REGISTERNOTIFY_CNF=> RCX_REGISTER_APP_REQ / RCX_REGISTER_APP_CNF See 2.1.18

0x198C/0x198D => 0x2F12/0x2F13	ECAT_ESM_UNREGISTERNOTIFY_REQ / ECAT_ESM_UNREGISTERNOTIFY_CNF => RCX_UNREGISTER_APP_REQ / RCX_UNREGISTER_APP_CNF See 2.1.19
0x1B18/0x1B19 => 0x1B18/0x1B19	ECAT_ESM_SET_QUEUE_CNF_AL_CONTROL_REQ / ECAT_ESM_SET_QUEUE_CNF_AL_CONTROL_CNF => ECAT_ESM_REGISTER_FOR_ALCONTROL_INDICATIONS_REQ / ECAT_ESM_REGISTER_FOR_ALCONTROL_INDICATIONS_CNF See 2.1.27
0x1B1A/0x1B1B => 0x1B1A/0x1B1B	ECAT_ESM_CLR_QUEUE_CNF_AL_CONTROL_REQ / ECAT_ESM_CLR_QUEUE_CNF_AL_CONTROL_CNF => ECAT_ESM_UNREGISTER_FROM_ALCONTROL_INDICATIONS_REQ / ECAT_ESM_UNREGISTER_FROM_ALCONTROL_INDICATIONS_CNF See 2.1.28
0x1B1C/0x1B1D => 0x1B1C/0x1B1D	ECAT_ESM_ALCONTROL_CHANGE_IND / ECAT_ESM_ALCONTROL_CHANGE_RES => ECAT_ESM_ALCONTROL_CHANGED_IND/ECAT_ESM_ALCONTROL_CHANGED_RES. See 2.1.29
0x1B1E/0x1B1F => 0x19DE /0x19DF	ECAT_ESM_ALSTATUS_CHANGE_REQ / ECAT_ESM_ALSTATUS_CHANGE_CNF=> ECAT_ESM_SET_ALSTATUS_REQ / ECAT_ESM_SET_ALSTATUS_CNF See 2.1.30
0x1B48/0x1B49 => 0x19DE/0x19DF	ECAT_ESM_SET_AL_STATUS_REQ / ECAT_ESM_SET_AL_STATUS_CNF => ECAT_ESM_SET_ALSTATUS_REQ / ECAT_ESM_SET_ALSTATUS_CNF See 2.1.31

Table 5: Replaced ESM Packets V2.5 =&gt; V4

### 2.1.3 ECAT\_ESM\_SII\_WRITE\_REQ / ECAT\_ESM\_SII\_WRITE\_CNF

This service still exists in V4.2.x. Compared to V2.5.16.0 the packet uses a data array of a fixed size of 1024 (ECAT\_ESM\_SII\_WRITE\_DATA\_BYTESIZE) bytes.

### 2.1.4 ECAT\_ESM\_SII\_READ\_REQ / ECAT\_ESM\_SII\_READ\_CNF

This service still exists in V4.2.x. Compared to V2.5.16.0 the packet uses a data array of a fixed size of 1024 (ECAT\_ESM\_SII\_READ\_DATA\_BYTESIZE) bytes.

### 2.1.5 ECAT\_ESM\_SII\_UPDATE\_VENDOR\_DATA\_IND / ECAT\_ESM\_SII\_UPDATE\_VENDOR\_DATA\_RES

This service does no longer exist in V4.2.x and is replaced by ECAT\_ESM\_SII\_WRITE\_IND / ECAT\_ESM\_SII\_WRITE\_RES.

The new packet works as follows:

- If a change in the Vendor ID has occurred ulSiiWriteStartAddress will have the value 0x10 and abData[2] will contain the new value for the Vendor ID.
- If a change in the Product Code has occurred ulSiiWriteStartAddress will have the value 0x14 and abData[2] will contain the new value for the Product Code.
- If a change in the Revision Number has occurred ulSiiWriteStartAddress will have the value 0x18 and abData[2] will contain the new value for the Revision Number.
- If a change in the Serial Number has occurred ulSiiWriteStartAddress will have the value 0x1C and abData[2] will contain the new value for the Serial Number.
-

## 2.1.6 **ECAT\_ESM\_SII\_WRITE\_VENDOR\_DATA\_REQ** /

## **ECAT\_ESM\_SII\_WRITE\_VENDOR\_DATA\_CNF**

This service does no longer exist in V4.2.x and is replaced by ECAT\_ESM\_SII\_WRITE\_REQ / ECAT\_ESM\_SII\_WRITE\_CNF.

## 2.1.7 **ECAT\_ESM\_ALCONTROL\_INIT\_IND** /

## **ECAT\_ESM\_ALCONTROL\_INIT\_RES**

This service does no longer exist in V4.2.x. The service ECAT\_ESM\_ALCONTROL\_CHANGED\_IND/ ECAT\_ESM\_ALCONTROL\_CHANGED\_RES provides the same functionality. Parameter tAlControl holds the EtherCAT state (uState).

## 2.1.8 **ECAT\_ESM\_ALCONTROL\_PRE\_OPERATIONAL\_IND** /

## **ECAT\_ESM\_ALCONTROL\_PRE\_OPERATIONAL\_RES**

This service does no longer exist in V4.2.x. The service ECAT\_ESM\_ALCONTROL\_CHANGED\_IND/ ECAT\_ESM\_ALCONTROL\_CHANGED\_RES provides the same functionality. Parameter tAlControl holds the EtherCAT state (uState).

## 2.1.9 **ECAT\_ESM\_ALCONTROL\_BOOTSTRAP\_IND** /

## **ECAT\_ESM\_ALCONTROL\_BOOTSTRAP\_RES**

This service does no longer exist in V4.2.x. The service ECAT\_ESM\_ALCONTROL\_CHANGED\_IND/ ECAT\_ESM\_ALCONTROL\_CHANGED\_RES provides the same functionality. Parameter tAlControl holds the EtherCAT state (uState).

## 2.1.10 **ECAT\_ESM\_ALCONTROL\_SAFE\_OPERATIONAL\_IND** /

## **ECAT\_ESM\_ALCONTROL\_SAFE\_OPERATIONAL\_RES**

This service does no longer exist in V4.2.x. The service ECAT\_ESM\_ALCONTROL\_CHANGED\_IND/ ECAT\_ESM\_ALCONTROL\_CHANGED\_RES provides the same functionality. Parameter tAlControl holds the EtherCAT state (uState).

## 2.1.11 **ECAT\_ESM\_ALCONTROL\_OPERATIONAL\_IND** /

## **ECAT\_ESM\_ALCONTROL\_OPERATIONAL\_RES**

This service does no longer exist in V4.2.x. The service ECAT\_ESM\_ALCONTROL\_CHANGED\_IND/ ECAT\_ESM\_ALCONTROL\_CHANGED\_RES provides the same functionality. Parameter tAlControl holds the EtherCAT state (uState).

## 2.1.12 **ECAT\_ESM\_ALSTATUS\_INIT\_IND** /

## **ECAT\_ESM\_ALSTATUS\_INIT\_RES**

This service does no longer exist in V4.2.x and is replaced by ECAT\_ESM\_ALSTATUS\_CHANGED\_IND / ECAT\_ESM\_ALSTATUS\_CHANGED\_RES.

Parameter tAlStatus of structure ECAT\_ESM\_ALSTATUS\_CHANGED\_IND\_DATA\_T holds the current EtherCAT state.



### **2.1.13 ECAT\_ESM\_ALSTATUS\_PRE\_OPERATIONAL\_IND / ECAT\_ESM\_ALSTATUS\_PRE\_OPERATIONAL\_RES**

This service does no longer exist in V4.2.x and is replaced by ECAT\_ESM\_ALSTATUS\_CHANGED\_IND / ECAT\_ESM\_ALSTATUS\_CHANGED\_RES.

Parameter tAlStatus of structure ECAT\_ESM\_ALSTATUS\_CHANGED\_IND\_DATA\_T holds the current EtherCAT state.

### **2.1.14 ECAT\_ESM\_ALSTATUS\_BOOTSTRAP\_IND / ECAT\_ESM\_ALSTATUS\_BOOTSTRAP\_RES**

This service does no longer exist in V4.2.x and is replaced by ECAT\_ESM\_ALSTATUS\_CHANGED\_IND / ECAT\_ESM\_ALSTATUS\_CHANGED\_RES.

Parameter tAlStatus of structure ECAT\_ESM\_ALSTATUS\_CHANGED\_IND\_DATA\_T holds the current EtherCAT state.

### **2.1.15 ECAT\_ESM\_ALSTATUS\_SAFE\_OPERATIONAL\_IND / ECAT\_ESM\_ALSTATUS\_SAFE\_OPERATIONAL\_RES**

This service does no longer exist in V4.2.x and is replaced by ECAT\_ESM\_ALSTATUS\_CHANGED\_IND / ECAT\_ESM\_ALSTATUS\_CHANGED\_RES.

Parameter tAlStatus of structure ECAT\_ESM\_ALSTATUS\_CHANGED\_IND\_DATA\_T holds the current EtherCAT state.

### **2.1.16 ECAT\_ESM\_ALSTATUS\_OPERATIONAL\_IND / ECAT\_ESM\_ALSTATUS\_OPERATIONAL\_RES**

This service does no longer exist in V4.2.x and is replaced by ECAT\_ESM\_ALSTATUS\_CHANGED\_IND / ECAT\_ESM\_ALSTATUS\_CHANGED\_RES.

Parameter tAlStatus of structure ECAT\_ESM\_ALSTATUS\_CHANGED\_IND\_DATA\_T holds the current EtherCAT state.

### **2.1.17 ECAT\_ESM\_SETINIT\_IND / ECAT\_ESM\_SETINIT\_RES**

This service still exists in V4.2.x and did not change in functionality compared to V2.5.16.0 but it was renamed to

ECAT\_ESM\_SETREADY\_REQ / ECAT\_ESM\_SETREADY\_CNF

### **2.1.18 ECAT\_ESM\_REGISTERNOTIFY\_REQ / ECAT\_ESM\_REGISTERNOTIFY\_CNF**

This service does no longer exist in V4.2.x. Only the RCX\_REGISTER\_APP\_REQ / RCX\_REGISTER\_APP\_CNF service shall be used in V4.2.x for registering an application for receiving general indications from the EtherCAT slave stack.

This service does no longer exist in V4.2.x. Only the RCX\_UNREGISTER\_APP\_REQ / RCX\_UNREGISTER\_APP\_CNF service shall be used in V4.2.x for unregistering an application from receiving general indications from the EtherCAT slave stack.

This service does no longer exist in V4.2.x.

This service does no longer exist in V4.2.x.

This service does no longer exist in V4.2.x.

This service does no longer exist in V4.2.x.

This service still exists in V4.2.x and did not change compared to V2.5.16.0.

This service does no longer exist in V4.2.x.

This service does no longer exist in V4.2.x.

### **2.1.27 ECAT\_ESM\_SET\_QUEUE\_CNF\_AL\_CONTROL\_REQ /** **ECAT\_ESM\_SET\_QUEUE\_CNF\_AL\_CONTROL\_CNF**

This service still exists in V4.2.x and did not change in functionality compared to V2.5.16.0 but it was renamed to

ECAT\_ESM\_REGISTER\_FOR\_ALCONTROL\_INDICATIONS\_REQ and  
ECAT\_ESM\_REGISTER\_FOR\_ALCONTROL\_INDICATIONS\_CNF.

### **2.1.28 ECAT\_ESM\_CLR\_QUEUE\_CNF\_AL\_CONTROL\_REQ /** **ECAT\_ESM\_CLR\_QUEUE\_CNF\_AL\_CONTROL\_CNF**

This service still exists in V4.2.x and did not change in functionality compared to V2.5.16.0 but it was renamed to

ECAT\_ESM\_UNREGISTER\_FROM\_ALCONTROL\_INDICATIONS\_REQ and  
ECAT\_ESM\_UNREGISTER\_FROM\_ALCONTROL\_INDICATIONS\_CNF.

### **2.1.29 ECAT\_ESM\_ALCONTROL\_CHANGE\_IND /** **ECAT\_ESM\_ALCONTROL\_CHANGE\_RES**

This service still exists in V4.2.x and did not change in functionality compared to V2.5.16.0 but it was renamed to

ECAT\_ESM\_ALCONTROL\_CHANGED\_IND and ECAT\_ESM\_ALCONTROL\_CHANGED\_RES.

### **2.1.30 ECAT\_ESM\_ALSTATUS\_CHANGE\_REQ /** **ECAT\_ESM\_ALSTATUS\_CHANGE\_CNF**

This service does no longer exist in V4.2.x. It is obsolete since ECAT\_ESM\_SET\_ALSTATUS\_REQ and ECAT\_ESM\_SET\_ALSTATUS\_CNF provide the same functionality.

### **2.1.31 ECAT\_ESM\_SET\_AL\_STATUS\_REQ /** **ECAT\_ESM\_SET\_AL\_STATUS\_CNF**

This service still exists in V4.2.x and did not change in functionality compared to V2.5.16.0 but it was renamed to

ECAT\_ESM\_SET\_ALSTATUS\_REQ and ECAT\_ESM\_SET\_ALSTATUS\_CNF.

### **2.1.32 ECAT\_ESM\_SET\_SAFEOP\_CHECK\_FN\_REQ /** **ECAT\_ESM\_SET\_SAFEOP\_CHECK\_FN\_CNF**

This service does no longer exist in V4.2.x.

### **2.1.33 ECAT\_ESM\_SET\_OP\_CHECK\_FN\_REQ /** **ECAT\_ESM\_SET\_OP\_CHECK\_FN\_CNF**

This service does no longer exist in V4.2.x.

## 2.2 DPM Packets

### 2.2.1 Replaced Packets

The following DPM packets of V2.5.16.0 have been replaced in V4.2.x:

Command Code	Packet Name
0x2CC4/0x2CC5=> 0x2CCE/0x2CCF	ECAT_DPM_WARMSTART_REQ / ECAT_DPM_WARMSTART_CNF=> ECAT_SET_CONFIG_REQ / ECAT_SET_CONFIG_CNF (Service still exists for legacy applications but is not recommended to be used)
0x2CCA/0x2CCB => 0x2CCE/0x2CCF	ECAT_DPM_SET_CONFIG_REQ / ECAT_DPM_SET_CONFIG_CNF=> ECAT_SET_CONFIG_REQ / ECAT_SET_CONFIG_CNF (Service still exists for legacy applications but is not recommended to be used)
0x2CCA/0x2CCB => 0x2CCE/0x2CCF	ECAT_DPM_SET_CONFIGURATION_REQ / ECAT_DPM_SET_CONFIGURATION_CNF=> ECAT_SET_CONFIG_REQ / ECAT_SET_CONFIG_CNF (Service still exists for legacy applications but is not recommended to be used)
0x2CCC/0x2CCD=> 0x2F34/0x2F35	ECAT_DPM_SET_UPDATE_CFG_REQ / ECAT_DPM_SET_UPDATE_CFG_CNF=> RCX_SET_HANDSHAKE_CONFIG_REQ/RCX_SET_HANDSHAKE_CONFIG_CNF.

Table 6: Replaced DPM Packets V2.5 => V4

### 2.2.2 ECAT\_DPM\_SET\_IO\_SIZE\_REQ ECAT\_DPM\_SET\_IO\_SIZE\_CNF

/

This service still exists in V4.2.x and did not change compared to V2.5.16.0.

### 2.2.3 ECAT\_DPM\_WARMSTART\_REQ ECAT\_DPM\_WARMSTART\_CNF

/

This service still exists in V4.2.x and did not change compared to V2.5.16.0. We urgently recommend to use this service only for legacy applications. New projects should use the new configuration capabilities of ECAT\_SET\_CONFIG\_REQ /CNF.

### 2.2.4 ECAT\_DPM\_SET\_STATION\_ALIAS\_REQ ECAT\_DPM\_SET\_STATION\_ALIAS\_CNF

/

This service still exists in V4.2.x and did not change compared to V2.5.16.0.

### 2.2.5 ECAT\_DPM\_GET\_STATION\_ALIAS\_REQ ECAT\_DPM\_GET\_STATION\_ALIAS\_CNF

/

This service still exists in V4.2.x and did not change compared to V2.5.16.0.

### 2.2.6 ECAT\_DPM\_SET\_CONFIG\_REQ ECAT\_DPM\_SET\_CONFIG\_CNF

/

This service still exists in V4.2.x and did not change compared to V2.5.16.0. We urgently recommend to use this service only for legacy applications. New projects should use the new configuration capabilities of ECAT\_SET\_CONFIG\_REQ /CNF.

This service still exists in V4.2.x and did not change compared to V2.5.16.0. We urgently recommend to use this service only for legacy applications. New projects should use the new configuration capabilities of ECAT\_SET\_CONFIG\_REQ /CNF.

This service does no longer exist in V4.2.x and is replaced by RCX\_SET\_HANDSHAKE\_CONFIG\_REQ and RCX\_SET\_HANDSHAKE\_CONFIG\_CNF.

## 2.3 CoE Packets

### 2.3.1 Removed Packets

The following CoE packets of V2.5 have been removed as the respective service does no longer exist in V4:

Command Code	Packet Name	Cause of Removal
0x19D0/ 0x19D1	ECAT_SDO_DOWNLOAD_EXP_REQ / ECAT_SDO_DOWNLOAD_EXP_CNF	replaced by ODV3 functionality
0x19D2/ 0x19D3	ECAT_SDO_UPLOAD_EXP_REQ / ECAT_SDO_UPLOAD_EXP_CNF	replaced by ODV3 functionality
0x1B26/ 0x1B27	ECAT_COE_SDO_SET_TIMEOUTS_REQ / ECAT_COE_SDO_SET_TIMEOUTS_CNF	replaced by ODV3 functionality
0x1B46/ 0x1B47	ECAT_COE_SDO_GET_TIMEOUTS_REQ / ECAT_COE_SDO_GET_TIMEOUTS_CNF	replaced by ODV3 functionality

Table 7: Removed CoE Packets V2.5 => V4

### 2.3.2 Replaced Packets

The following CoE packets of V2.5 have been replaced in V4:

Command Code	Packet Name
0x199C/0x199D 0x6A02/0x6A03	ECAT_LOCAL_SDO_DOWNLOAD_EXP_REQ / ECAT_LOCAL_SDO_DOWNLOAD_EXP_CNF=> ODV3_WRITE_OBJECT_REQ/ ODV3_WRITE_OBJECT_CNF See 2.3.4
0x199E/0x199F 0x6A00/0x6A01	ECAT_LOCAL_SDO_UPLOAD_EXP_REQ / ECAT_LOCAL_SDO_UPLOAD_EXP_CNF=> ODV3_READ_OBJECT_REQ/ ODV3_READ_OBJECT_CNF See 2.3.5

Table 8: Replaced CoE Packets V2.5 => V4

### 2.3.3 ECAT\_COE\_SEND\_EMERGENCY\_REQ / ECAT\_COE\_SEND\_EMERGENCY\_CNF

This service still exists in V4.2.x and did not change compared to V2.5.16.0.

### 2.3.4 ECAT\_LOCAL\_SDO\_DOWNLOAD\_EXP\_REQ / ECAT\_LOCAL\_SDO\_DOWNLOAD\_EXP\_CNF

This service does no longer exist in V4.2.x and is replaced by ODV3\_WRITE\_OBJECT\_REQ/ ODV3\_WRITE\_OBJECT\_CNF.

The following differences apply to the request packet:

- There is no equivalent to the station address parameter (ulServerAddress).
- The index parameter is a 16 bit value.
- The subindex parameter is a 8 bit value.
- The data area is only 1024 bytes long, not 1536 bytes.
- "Simple variable" objects (object code 0x0007) are addressed by bSubIndex = 0.

The following differences apply to the confirmation packet:

- The index parameter is now returned (16 bit).
- The subindex parameter is now returned (8 bit).

## /

The following differences apply to the request packet:

- There is no equivalent to the station address parameter (ulServerAddress).
- The index parameter is a 16 bit value.
- The subindex parameter is a 8 bit value.
- "Simple variable" objects (object code 0x0007) are addressed by bSubIndex = 0.

The following differences apply to the confirmation packet:

- The index parameter is now returned additionally (16 bit).
- The subindex parameter is now returned additionally (8 bit).
- The maximum segment length parameter is now returned additionally.
- The total data bytes parameter is now returned additionally.
- The data area is only 1024 bytes long, not 1556 bytes.

## /

This service does no longer exist in V4.2.x.

## /

This service does no longer exist in V4.2.x.

## /

This service does no longer exist in V4.2.x.

## /

This service does no longer exist in V4.2.x.

## 2.4 OD Packets

### 2.4.1 Removed Packets

The following OD packets of V2.5.x have been removed as the respective service does no longer exist in V4.2.x:

Command Code	Packet Name	Cause of Removal
0x19C8/ 0x19C9	ECAT_OD_SET_INDICATION_TIMEOUT_REQ / ECAT_OD_SET_INDICATION_TIMEOUT_CNF	replaced by ODV3 functionality
0x19CA/ 0x19CB	ECAT_OD_GET_INDICATION_TIMEOUT_REQ / ECAT_OD_GET_INDICATION_TIMEOUT_CNF	replaced by ODV3 functionality
0x1B14/ 0x1B15	ECAT_OD_NOTIFY_READ_IND / ECAT_OD_NOTIFY_READ_RES	replaced by ODV3 functionality
0x1B16/ 0x1B17	ECAT_OD_NOTIFY_WRITE_IND / ECAT_OD_NOTIFY_WRITE_RES	replaced by ODV3 functionality
0x1B24/ 0x1B25	ECAT_OD_UNDEFINED_READ_PREPARE_IND / ECAT_OD_UNDEFINED_READ_PREPARE_RES	replaced by ODV3 functionality
0x1B26/ 0x1B27	ECAT_OD_UNDEFINED_READ_DATA_IND / ECAT_OD_UNDEFINED_READ_DATA_RES	replaced by ODV3 functionality
0x1B28/ 0x1B29	ECAT_OD_UNDEFINED_WRITE_DATA_IND / ECAT_OD_UNDEFINED_WRITE_DATA_RES	replaced by ODV3 functionality
0x1B30/ 0x1B31	ECAT_OD_SDOINFO_REGISTER_REQ / ECAT_OD_SDOINFO_REGISTER_CNF	replaced by ODV3 functionality
0x1B32/ 0x1B33	ECAT_OD_SDOINFO_UNREGISTER_REQ / ECAT_OD_SDOINFO_UNREGISTER_CNF	replaced by ODV3 functionality
0x1B3A / 0x1B3B	ECAT_OD_NOTIFY_REGISTER_RESET_REQ / ECAT_OD_NOTIFY_REGISTER_RESET_CNF	replaced by ODV3 functionality

Table 9: Removed OD Packets V2.5 => V4

### 2.4.2 Replaced Packets

The following OD packets of V2.5 have been replaced in V4:

Command Code	Packet Name
0x1B00/0x1B01=> 0x6A80/0x6A81	ECAT_OD_CREATE_OBJECT_REQ / ECAT_OD_CREATE_OBJECT_CNF=> ODV3_CREATE_OBJECT_REQ/ ODV3_CREATE_OBJECT_CNF See 2.4.5
0x1B02/0x1B03=> 0x6A82/0x6A83	ECAT_OD_CREATE_SUBOBJECT_REQ / ECAT_OD_CREATE_SUBOBJECT_CNF=> ODV3_CREATE_SUBOBJECT_REQ/ ODV3_CREATE_SUBOBJECT_CNF See 2.4.6
0x1B04/0x1B05=> 0x6A84/0x6A85	ECAT_OD_DELETE_OBJECT_REQ / ECAT_OD_DELETE_OBJECT_CNF => ODV3_DELETE_OBJECT_REQ/ ODV3_DELETE_OBJECT_CNF Only applicable in case fDeleteWholeObject = TRUE. See 2.4.7
0x1B04/0x1B05=> 0x6A86/0x6A87	ECAT_OD_DELETE_OBJECT_REQ / ECAT_OD_DELETE_OBJECT_CNF => ODV3_DELETE_SUBOBJECT_REQ/ ODV3_DELETE_SUBOBJECT_CNF Only applicable in case fDeleteWholeObject = FALSE See 2.4.7
0x1B06/0x1B07=> 0x6AC0/0x6AC1	ECAT_OD_CREATE_DATATYPE_REQ / ECAT_OD_CREATE_DATATYPE_CNF => ODV3_CREATE_DATATYPE_REQ/ ODV3_CREATE_DATATYPE_CNF See 2.4.8
0x1B08/0x1B09=> 0x6AC2/0x6AC3	ECAT_OD_DELETE_DATATYPE_REQ / ECAT_OD_DELETE_DATATYPE_CNF => ODV3_DELETE_DATATYPE_REQ/ ODV3_DELETE_DATATYPE_CNF See 2.4.9
0x1B10/0x1B11=> 0x6A90/0x6A91	ECAT_OD_NOTIFY_REGISTER_REQ / ECAT_OD_NOTIFY_REGISTER_CNF => ODV3_REGISTER_OBJECT_NOTIFY_REQ/ ODV3_REGISTER_OBJECT_NOTIFY_CNF Only applicable for notifications on objects. See 2.4.10



0x1B10/0x1B11=> 0x6A94/0x6A95	ECAT_OD_NOTIFY_REGISTER_REQ / ECAT_OD_NOTIFY_REGISTER_CNF => ODV3_REGISTER_SUBOBJECT_NOTIFY_REQ/ODV3_REGISTER_SUBOBJECT_NOTIFY_CNF Only applicable for notifications on subobjects. See 2.4.10
0x1B12/0x1B13=> 0x6A92/0x6A93	ECAT_OD_NOTIFY_UNREGISTER_REQ / ECAT_OD_NOTIFY_UNREGISTER_CNF => ODV3_UNREGISTER_OBJECT_NOTIFY_REQ/ ODV3_UNREGISTER_OBJECT_NOTIFY_CNF Only applicable for notifications on objects. See 2.4.11
0x1B12/0x1B13=> 0x6A96/0x6A97	ECAT_OD_NOTIFY_UNREGISTER_REQ / ECAT_OD_NOTIFY_UNREGISTER_CNF => ODV3_UNREGISTER_SUBOBJECT_NOTIFY_REQ/ODV3_UNREGISTER_SUBOBJECT_NOTIFY_CN F Only applicable for notifications on subobjects. See 2.4.11
0x1B20/0x1B21=> 0x6AA0/0x6AA1	ECAT_OD_UNDEFINED_NOTIFY_REGISTER_REQ / ECAT_OD_UNDEFINED_NOTIFY_REGISTER_CNF=> ODV3_REGISTER_UNDEFINED_NOTIFY_REQ/ ODV3_REGISTER_UNDEFINED_NOTIFY_CNF See 2.4.14
0x1B22/ 0x1B23=> 0x6AA2/0x6AA3	ECAT_OD_UNDEFINED_NOTIFY_UNREGISTER_REQ / ECAT_OD_UNDEFINED_NOTIFY_UNREGISTER_CNF=> ODV3_UNREGISTER_UNDEFINED_NOTIFY_REQ/ ODV3_UNREGISTER_UNDEFINED_NOTIFY_CNF
0x1B34/0x1B35=> 0x6A10/0x6A11	ECAT_OD_SDOINFO_GET_LIST_IND / ECAT_OD_SDOINFO_GET_LIST_RES=> ODV3_GET_OBJECT_LIST_IND/ ODV3_GET_OBJECT_LIST_RES. See 2.4.21
0x1B36/0x1B37=> 0x6A12/0x6A13	ECAT_OD_SDOINFO_GET_OBJ_DESC_IND / ECAT_OD_SDOINFO_GET_OBJ_DESC_RES=> ODV3_GET_OBJECT_INFO_IND/ ODV3_GET_OBJECT_INFO_RES See 2.4.22
0x1B38/0x1B39=> 0x6A14/0x6A15	ECAT_OD_SDOINFO_GET_ENTRY_DESC_IND / ECAT_OD_SDOINFO_GET_ENTRY_DESC_RES=> ODV3_GET_SUBOBJECT_INFO_IND / ODV3_GET_SUBOBJECT_INFO_RES See 2.4.23
0x1B3C/0x1B3D=> 0x6AB4/0x6AB5	ECAT_OD_SET_OBJECT_NAME_REQ / ECAT_OD_SET_OBJECT_NAME_CNF => ODV3_SET_OBJECT_NAME_REQ/ ODV3_SET_OBJECT_NAME_CNF See 2.4.25
0x1B3E/0x1B3F=> 0x6AB6/0x6AB7	ECAT_OD_SET_SUBOBJECT_NAME_REQ / ECAT_OD_SET_SUBOBJECT_NAME_CNF => ODV3_SET_SUBOBJECT_NAME_REQ/ ODV3_SET_SUBOBJECT_NAME_CNF See 2.4.26

Table 10: Replaced OD Packets V2.5 =&gt; V4

### 2.4.3 ECAT\_OD\_SET\_INDICATION\_TIMEOUT\_REQ / ECAT\_OD\_SET\_INDICATION\_TIMEOUT\_CNF

This service does no longer exist in V4.2.x.

### 2.4.4 ECAT\_OD\_GET\_INDICATION\_TIMEOUT\_REQ / ECAT\_OD\_GET\_INDICATION\_TIMEOUT\_CNF

This service does no longer exist in V4.2.x.

### 2.4.5 ECAT\_OD\_CREATE\_OBJECT\_REQ / ECAT\_OD\_CREATE\_OBJECT\_CNF

This service does no longer exist in V4.2.x and is replaced by ODV3\_CREATE\_OBJECT\_REQ/ ODV3\_CREATE\_OBJECT\_CNF.

The index parameter usIndex remains unchanged.

The following differences apply to the request packet:

- The parameter `bMaxNumOfSubObjs` contains the maximum number of subobjects in the object container to be created. It replaces the former parameter `bMaxSubObjs`. The parameter `bNumSubObjs` is not required anymore.
- The parameter `bObjectCode` is coded in the same way as in the EtherCAT V2.5 stack
- The parameter `usAccessFlags` is new in the EtherCAT V4.2.x stack. It describes to what lists an object belongs. It may not be confused with the parameter `usAccessRights`! See table “Object Access Flags during Object Creation” in the ODV3 Protocol API Manual (reference 0).
- The parameter `bValueInfo` is new in the EtherCAT V4.2.x stack. It specifies which fields are provided in `abData[1024]`, see below. See table “Value Info Flags during Object Creation” in the ODV3 Protocol API Manual (reference 0).
- The parameter `blndicationFlags` is new in the EtherCAT V4.2.x stack. It controls whether specific indication options will be registered immediately on object creation. The following options exist for indication registration
  - `ODV3_INDICATION_FLAGS_ON_READ` (Bit 0)
  - `ODV3_INDICATION_FLAGS_ON_WRITE` (Bit 1)
  - `ODV3_INDICATION_FLAGS_ON_RW_UNDEFINED_SUBOBJ`
  - `ODV3_INDICATION_FLAGS_ON_INFO_UNDEFINED_SUBOBJ`
  - `ODV3_INDICATION_FLAGS_ON_WRITE_INVALIDATED`

For a detailed description, see table “Indication Flags during Object Creation” in the ODV3 Protocol API Manual (reference 0).

- The parameter `usDatatype` specifies the actual data type to be set on the object. If a variable (VAR) is created, the subindex 0 will be assigned with the same data type.  
If an object with subobjects is created, the data type is only assigned to the object. The data type is not passed into any subobject.

The following two parameters are only evaluated on simple variables (VAR).

- The parameter `usAccessRights` replaces the former parameter `usObjAccess`. It may not be confused with the parameter `usAccessFlags` described above! The coding has been extended by bits D6 to D9. Use the EtherCAT table in section “Access rights” of the ODV3 Protocol API Manual (reference 0).
- The parameter `ulMaxFieldUnits` specifies the maximum number of data type units subindex 0 can contain if created. See reference 0.
- The parameter `ulTotalDataBytes` refers to the entire data transfers i.e. the summed data length of all data fragments (this is used for fragmentation purposes)
- The parameter field `abData[1024]` contains the additional data to be used for creation. The content is controlled by `bValueInfo`.

The following differences apply to the confirmation packet:

- The index parameter `usIndex` is now returned additionally (same value as in request packet).

## 2.4.6 ECAT\_OD\_CREATE\_SUBOBJECT\_REQ ECAT\_OD\_CREATE\_SUBOBJECT\_CNF

This service does no longer exist in V4.2.x and is replaced by ODV3\_CREATE\_SUBOBJECT\_REQ/ ODV3\_CREATE\_SUBOBJECT\_CNF.

The index parameter usIndex remains unchanged.

The following differences apply to the request packet:

- The subindex parameter bSubIndex replaces the former parameter bSubIdx.
- The parameter bValueInfo is new in the EtherCAT V4.2.x stack. It specifies which fields are provided in abData[1024], see below. See table “Value Info Flags during Object Creation” in the ODV3 Protocol API Manual (reference 0).
- The parameter bIndicationFlags is new in the EtherCAT V4.2.x stack. It controls whether specific indication options will be registered immediately on subobject creation. The following options exist for indication registration
  - ODV3\_INDICATION\_FLAGS\_ON\_READ (Bit 0)
  - ODV3\_INDICATION\_FLAGS\_ON\_WRITE (Bit 1)
  - ODV3\_INDICATION\_FLAGS\_ON\_RW\_UNDEFINED\_SUBOBJ
  - ODV3\_INDICATION\_FLAGS\_ON\_INFO\_UNDEFINED\_SUBOBJ
  - ODV3\_INDICATION\_FLAGS\_ON\_WRITE\_INVALIDATED

For a detailed description, see table “Indication Flags during Object Creation” in the ODV3 Protocol API Manual (reference 0).

- The parameter usAccessRights replaces the former parameter usSubObjAccess. It may not be confused with the parameter usAccessFlags described above! The coding has been extended by bits D6 to D9. Use the EtherCAT table in section “Access rights” of the ODV3 Protocol API Manual (reference 0).
- The parameter usDatatype specifies the actual data type to be set on the object. If a variable (VAR) is created, the subindex 0 will be assigned with the same data type.
- The parameter ulMaxFieldUnits specifies the maximum number of data type units subindex 0 can contain if created. See reference 0.
- The parameter ulTotalDataBytes has been added. It refers to the entire data transfers i.e. the summed data length of all data fragments. (This is used for fragmentation purposes)
- The parameter field abData[1024] contains the additional data to be used for creation. The content is controlled by bValueInfo.
- For „simple variable“ objects (object code 0x0007) bSubIndex = 0 shall be used for the creation.

The following differences apply to the confirmation packet:

- The index parameter usIndex is now returned additionally (same value as in request packet).
- The subindex parameter bSubIndex is now returned additionally (same value as in request packet).

## **2.4.7      ECAT\_OD\_DELETE\_OBJECT\_REQ              ECAT\_OD\_DELETE\_OBJECT\_CNF**

/

This service does no longer exist in V4.2.x. It has been used to delete both objects and subobjects within the object dictionary. It is now replaced by the two services ODV3\_DELETE\_OBJECT\_REQ/ ODV3\_DELETE\_OBJECT\_CNF for deleting objects (corresponding to the former case fDeleteWholeObject = TRUE) and ODV3\_DELETE\_SUBOBJECT\_REQ/ ODV3\_DELETE\_SUBOBJECT\_CNF for deleting subobjects (corresponding to the former case fDeleteWholeObject = FALSE).

The index parameter usIndex remains unchanged.

The following differences apply to the request packet:

- The subindex parameter bSubIndex replaces the former parameter bSubIdx. (only for ODV3\_DELETE\_SUBOBJECT\_REQ/ ODV3\_DELETE\_SUBOBJECT\_CNF)
- The parameters fDeleteWholeObject and hSender are no longer present.
- For „simple variable“ objects (object code 0x0007) bSubIndex = 0 shall be used for the creation.

The following differences apply to the confirmation packet:

- The index parameter usIndex is now returned additionally (same value as in request packet).
- The subindex parameter bSubIndex is now returned additionally (only for ODV3\_DELETE\_SUBOBJECT\_REQ/ ODV3\_DELETE\_SUBOBJECT\_CNF, same value as in request packet).

## **2.4.8      ECAT\_OD\_CREATE\_DATATYPE\_REQ              ECAT\_OD\_CREATE\_DATATYPE\_CNF**

/

This service does no longer exist in V4.2.x and is replaced by ODV3\_CREATE\_DATATYPE\_REQ/ ODV3\_CREATE\_DATATYPE\_CNF.

The datatype parameter usDatatype remains unchanged.

The following differences apply to the request packet:

- The datatype bit length parameter is now renamed to ulDatatypeBitLength.
- The boolean parameter fVariableLength is not available anymore in the EtherCAT V4.2.x stack.

The following differences apply to the confirmation packet:

- The datatype parameter usDatatype is now returned additionally (same value as in request packet).

## **2.4.9      ECAT\_OD\_DELETE\_DATATYPE\_REQ              ECAT\_OD\_DELETE\_DATATYPE\_CNF**

/

This service does no longer exist in V4.2.x and is replaced by ODV3\_DELETE\_DATATYPE\_REQ/ ODV3\_DELETE\_DATATYPE\_CNF.

The datatype parameter usDatatype remains unchanged.

The following differences apply to the confirmation packet:

- The datatype parameter usDatatype is now returned additionally (same value as in request packet).

## **2.4.10 ECAT\_OD\_NOTIFY\_REGISTER\_REQ /** **ECAT\_OD\_NOTIFY\_REGISTER\_CNF**

This service does no longer exist in V4.2.x and is replaced by ODV3\_REGISTER\_OBJECT\_NOTIFY\_REQ/ ODV3\_REGISTER\_OBJECT\_NOTIFY\_CNF and by ODV3\_REGISTER\_SUBOBJECT\_NOTIFY\_REQ/ ODV3\_REGISTER\_SUBOBJECT\_NOTIFY\_CNF.

The index parameter usIndex remains unchanged.

The following differences apply to the request packet:

- The boolean parameters fReadNotify and fWriteNotify are both replaced by bIndicationFlags. bIndicationFlags offers additional options.
- The subindex parameter bSubIndex allows notifications on subobjects (only for ODV3\_REGISTER\_SUBOBJECT\_NOTIFY\_CNF).

The following differences apply to the confirmation packet:

- The index parameter usIndex is now returned additionally (same value as in request packet).
- The subindex parameter bSubIndex is now returned additionally (only for ODV3\_REGISTER\_SUBOBJECT\_NOTIFY\_CNF, same value as in request packet).

## **2.4.11 ECAT\_OD\_NOTIFY\_UNREGISTER\_REQ /** **ECAT\_OD\_NOTIFY\_UNREGISTER\_CNF**

This service does no longer exist in V4.2.x and is replaced by ODV3\_UNREGISTER\_OBJECT\_NOTIFY\_REQ/ ODV3\_UNREGISTER\_OBJECT\_NOTIFY\_CNF and by ODV3\_UNREGISTER\_SUBOBJECT\_NOTIFY\_REQ/ ODV3\_UNREGISTER\_SUBOBJECT\_NOTIFY\_CNF.

The index parameter usIndex remains unchanged.

The following differences apply to the request packet:

- The boolean parameters fReadNotify and fWriteNotify are no longer available.
- The subindex parameter bSubIndex allows notifications on subobjects (only for ODV3\_UNREGISTER\_SUBOBJECT\_NOTIFY\_CNF).

The following differences apply to the confirmation packet:

- The index parameter usIndex is now returned additionally (same value as in request packet).

## **2.4.12 ECAT\_OD\_NOTIFY\_READ\_IND /** **ECAT\_OD\_NOTIFY\_READ\_RES**

This service does no longer exist in V4.2.x and is replaced by ODV3\_WRITE\_OBJECT\_IND/RES.

## **2.4.13 ECAT\_OD\_NOTIFY\_WRITE\_IND /** **ECAT\_OD\_NOTIFY\_WRITE\_RES**

This service does no longer exist in V4.2.x and is replaced by ODV3\_READ\_OBJECT\_IND/RES.

## **2.4.14 ECAT\_OD\_UNDEFINED\_NOTIFY\_REGISTER\_REQ /** **ECAT\_OD\_UNDEFINED\_NOTIFY\_REGISTER\_CNF**

This service does no longer exist in V4.2.x and is replaced by ODV3\_REGISTER\_UNDEFINED\_NOTIFY\_REQ/CNF.

Neither request nor confirmation packet have any parameters. This remains unchanged in V4.2.x.

## **2.4.15 ECAT\_OD\_UNDEFINED\_NOTIFY\_UNREGISTER\_REQ /** **ECAT\_OD\_UNDEFINED\_NOTIFY\_UNREGISTER\_CNF**

This service does no longer exist in V4.2.x and is replaced by ODV3\_UNREGISTER\_UNDEFINED\_NOTIFY\_REQ/CNF.

Neither request nor confirmation packet have any parameters. This remains unchanged in V4.2.x.

## **2.4.16 ECAT\_OD\_UNDEFINED\_READ\_PREPARE\_IND /** **ECAT\_OD\_UNDEFINED\_READ\_PREPARE\_RES**

This service does no longer exist in V4.2.x and is replaced by ODV3\_GET\_OBJECT\_SIZE\_IND/RES.

- A subobject of a "simple variable" object (object code 0x0007) is addressed by bSubIndex = 0.

## **2.4.17 ECAT\_OD\_UNDEFINED\_READ\_DATA\_IND /** **ECAT\_OD\_UNDEFINED\_READ\_DATA\_RES**

This service does no longer exist in V4.2.x and is replaced by ODV3\_READ\_OBJECT\_IND/RES.

- A subobject of a "simple variable" object (object code 0x0007) is addressed by bSubIndex = 0.

## **2.4.18 ECAT\_OD\_UNDEFINED\_WRITE\_DATA\_IND /** **ECAT\_OD\_UNDEFINED\_WRITE\_DATA\_RES**

This service does no longer exist in V4.2.x and is replaced by ODV3\_WRITE\_OBJECT\_IND/RES.

- A subobject of a "simple variable" object (object code 0x0007) is addressed by bSubIndex = 0..

## **2.4.19 ECAT\_OD\_SDOINFO\_REGISTER\_REQ /** **ECAT\_OD\_SDOINFO\_REGISTER\_CNF**

This service does no longer exist in V4.2.x and is replaced by ODV3\_REGISTER\_OBJINFO\_NOTIFY\_REQ/CNF.

## **2.4.20 ECAT\_OD\_SDOINFO\_UNREGISTER\_REQ /** **ECAT\_OD\_SDOINFO\_UNREGISTER\_CNF**

This service does no longer exist in V4.2.x and is replaced by ODV3\_UNREGISTER\_OBJINFO\_NOTIFY\_REQ/CNF.

## 2.4.21 ECAT\_OD\_SDOINFO\_GET\_LIST\_IND ECAT\_OD\_SDOINFO\_GET\_LIST\_RES

This service does no longer exist in V4.2.x and is replaced by ODV3\_GET\_OBJECT\_LIST\_IND/RES.

The following differences apply to the indication packet:

- The parameter usListType for selecting which type of objects to list is replaced by a similar new parameter usObjAccessMask in the EtherCAT V4.2.x stack.

The relationship between these parameters is as follows:

Description	Value usListType	Value usObjAccessMask
Objects which are used to store settings These objects hold parameters which can be written during startup	ECAT_COE_OBJLIST_CONFIG_DATA (0x0005)	ODV3_ACCESS_FLAGS_SETTINGS (0x8000) Bit 15 set
Objects which are required to be stored for device replacement	ECAT_COE_OBJLIST_BACKUP (0x0004)	ODV3_ACCESS_FLAGS_BACKUP (0x4000) Bit 14 set
Objects which can be mapped into TxPDOs	ECAT_COE_OBJLIST_TXPDO_MAPPABLE (0x0003)	ODV3_ACCESS_FLAGS_TXPDO_MAPPABLE (0x2000) Bit 13 set
Objects which can be mapped into RxPDOs	ECAT_COE_OBJLIST_RXPDO_MAPPABLE (0x0002)	ODV3_ACCESS_FLAGS_RXPDO_MAPPABLE (0x1000) Bit 12 set
All objects	ECAT_COE_OBJLIST_ALL (0x0001)	-
	ECAT_COE_OBJLIST_LENGTH (0x)	-
Force indexed (only on indexed objects, uses simple var bValueInfo fields)		ODV3_ACCESS_FLAGS_FORCE_INDEXED (0x0800)
Create subindex 0		ODV3_ACCESS_FLAGS_CREATE_SUBINDEX_0 (0x0400)

Table 11: Relationship between Parameter usListType of packet ECAT\_OD\_SDOINFO\_GET\_LIST\_IND and Parameter usObjAccessMask of packet ODV3\_GET\_OBJECT\_LIST\_IND

- The parameter usObjAccessCompare is new in EtherCAT slave stack V4.2.x. For more information, see reference 0.

The following differences apply to the response packet:

- The mask parameter usObjAccessMask is now returned additionally (same value as in request packet).
- The comparison parameter usObjAccessCompare is now returned additionally (same value as in request packet).
- The parameter uiTotalDataBytes has been added. It refers to the entire data transfers i.e. the summed data length of all data fragments. (This is used for fragmentation purposes)
- In the EtherCAT Slave stack V4.2.x, the parameter field ausIndex[] is dimensioned for 512 index entries instead of 1024.

## 2.4.22 **ECAT\_OD\_SDOINFO\_GET\_OBJ\_DESC\_IND** / **ECAT\_OD\_SDOINFO\_GET\_OBJ\_DESC\_RES**

This service does no longer exist in V4.2.x and is replaced by ODV3\_GET\_OBJECT\_INFO\_IND/ODV3\_GET\_OBJECT\_INFO\_RES.

The index parameter usIndex of the indication packet remains unchanged.

The following differences apply to the response packet:

- The index parameter usIndex must now be returned additionally (just use the same value as in the indication packet).
- The parameter usDatatype specifies the actual data type to be set on the object. If a variable (VAR) is created, the subindex 0 will be assigned with the same data type.  
If an object with subobjects is created, the data type is only assigned to the object. The data type is not passed into any subobject.
- The parameter usAccessFlags is new in the EtherCAT V4.2.x stack. It describes to what lists an object belongs. It may not be confused with the parameter usAccessRights! See table "Object Access Flags during Object Creation" in the ODV3 Protocol API Manual (reference 0).
- The parameter bMaxNumOfSubObjs contains the maximum number of subobjects in the object container to be created. It replaces the former parameter bMaxSubObjs. The parameter bNumSubObjs is not required anymore.
- The parameter bObjectCode is coded in the same way as in the EtherCAT V2.5 stack
- The parameter ulTotalDataBytes has been added. It refers to the entire data transfers i.e. the summed data length of all data fragments. (This is used for fragmentation purposes)
- The parameter field abName[1024] replaces the former parameter field szName[128] (the name of object).

## 2.4.23 **ECAT\_OD\_SDOINFO\_GET\_ENTRY\_DESC\_IND** / **ECAT\_OD\_SDOINFO\_GET\_ENTRY\_DESC\_RES**

This service does no longer exist in V4.2.x and is replaced by ODV3\_GET\_SUBOBJECT\_INFO\_IND / ODV3\_GET\_SUBOBJECT\_INFO\_RES.

The index parameter usIndex of the indication packet remains unchanged.

The following differences apply to the indication packet:

- The subindex parameter bSubIndex replaces the former parameter bSubIdx.
- The parameter bRequestedValueInfo replaces the former parameter bValueInfo.

The following differences apply to the response packet:

- The index parameter usIndex and the subindex parameter bSubIndex must now be returned additionally (just use the same value as in the indication packet).
- The parameter bValueInfo specifies which fields are provided in abData[1024], see below. See table "Value Info Flags during Object Creation" in the ODV3 Protocol API Manual (reference 0).
- The parameter ulTotalDataBytes has been added. It refers to the entire data transfers i.e. the summed data length of all data fragments. (This is used for fragmentation purposes)
- The parameter usAccessRights accepts the same values as in the former stack. Additionally, the following access rights are available in the EtherCAT V4.2.x stack:



Bit	Full Name	Description
D15	ECAT_OD_WRITE_INIT	Write Access in Initial State
D14	ECAT_OD_READ_INIT	Read Access in Initial State

Table 12: Allowed values for Parameter *usAccessRights*

- The parameter *usDatatype* specifies the actual data type to be set on the object. If a variable (VAR) is created, the subindex 0 will be assigned with the same data type. For a list of data types refer to the ODV3 Protocol API Manual (reference 0).
- The parameter *ulDataBitLen* replaces the former parameter *usBitLength*. It contains the length of the subobject in bits.
- The parameter *usFieldLen* has been added. It must contain the field length (i.e. the number of data type units)
- The parameter field *abData[1024]* (containing the description of the object) is now dimensioned for 1024 bytes. Formerly the actually usable length depended on the mailbox size and was limited at 2048 bytes.

#### 2.4.24 ECAT\_OD\_NOTIFY\_REGISTER\_RESET\_REQ / ECAT\_OD\_NOTIFY\_REGISTER\_RESET\_CNF

This service does no longer exist in V4.2.x.

#### 2.4.25 ECAT\_OD\_SET\_OBJECT\_NAME\_REQ / ECAT\_OD\_SET\_OBJECT\_NAME\_CNF

This service does no longer exist in V4.2.x and is replaced by ODV3\_SET\_OBJECT\_NAME\_REQ/ ODV3\_SET\_OBJECT\_NAME\_CNF.

The index parameter *usIndex* remains unchanged.

The following differences apply to the request packet:

- In the EtherCAT Slave stack V4.2.x, the parameter field *szName[]* is dimensioned for 200 characters entries instead of 128.

The following differences apply to the confirmation packet:

- The index parameter *usIndex* is now returned additionally (same value as in request packet).

#### 2.4.26 ECAT\_OD\_SET\_SUBOBJECT\_NAME\_REQ / ECAT\_OD\_SET\_SUBOBJECT\_NAME\_CNF

This service does no longer exist in V4.2.x and is replaced by ODV3\_SET\_SUBOBJECT\_NAME\_REQ/ ODV3\_SET\_SUBOBJECT\_NAME\_CNF.

The index parameter *usIndex* remains unchanged.

The following differences apply to the request packet:

- The subindex parameter *bSubIndex* replaces the former parameter *bSubIdx*.
- In the EtherCAT Slave stack V4.2.x, the parameter field *szName[]* is dimensioned for 200 characters entries instead of 128.

The following differences apply to the confirmation packet:

- The index parameter *usIndex* is now returned additionally (same value as in request packet).
- The subindex parameter *bSubIndex* is now returned additionally (same value as in request packet).

## 3 New Packets

### 3.1 General Packets

#### 3.1.1 ECAT\_SET\_CONFIG\_REQ / ECAT\_SET\_CONFIG\_CNF

This service has a new concept compared to V2.5.x, see Protocol API Manual (Reference 0).

We urgently recommend to use this service for new applications. For legacy applications the old configuration packets are still available.

### 3.2 ESM Packets

#### 3.2.1 ECAT\_ESM\_REGISTER\_FOR\_SIIWRITE\_INDICATIONS\_REQ / ECAT\_ESM\_REGISTER\_FOR\_SIIWRITE\_INDICATIONS\_CNF

This service is used to register for SII write indications. For further information see Protocol API Manual (Reference 0).

#### 3.2.2 ECAT\_ESM\_UNREGISTER\_FROM\_SIIWRITE\_INDICATIONS\_REQ / ECAT\_ESM\_UNREGISTER\_FROM\_SIIWRITE\_INDICATIONS\_CNF

This service is used to unregister from SII write indications. For further information see Protocol API Manual (Reference 0).

#### 3.2.3 ECAT\_ESM\_SII\_WRITE\_IND / ECAT\_ESM\_SII\_WRITE\_RES

This service is used to indicate SII write indications. For further information see Protocol API Manual (Reference 0).

#### 3.2.4 ECAT\_ESM\_ALSTATUS\_CHANGED\_IND / ECAT\_ESM\_ALSTATUS\_CHANGED\_RES

This service is used to indicate AL status changes. For further information see Protocol API Manual (Reference 0).

#### 3.2.5 ECAT\_ESM\_GET\_ALSTATUS\_REQ / ECAT\_ESM\_GET\_ALSTATUS\_CNF

This service is used to read the AL status. For further information see Protocol API Manual (Reference 0).

## 4 Changes in Error Codes

The EtherCAT Slave stack V4.2.x uses its own, completely different set of error codes and diag codes compared to the former EtherCAT Slave stack V2.5. Depending on the component of the stack causing the error/ diag message, the following high bytes of the codes will be used:

ESM	0xC0AF
MBX	0xC0B0
CoE	0xC0B1
EoE	0xC0B2
FoE	0xC0B3
SoE	0xC0B4
Diag/ General	0x80AF/0x00AF
Diag/ DPM	0xC0AE
ODV3	0xC09B

Table 13: Error/diagnosis messages

## 5 Appendix

### 5.1 List of Tables

Table 1: List of Revisions .....	4
Table 2: References .....	4
Table 3: Removed ESM Packets V2.5 => V4.....	5
Table 4: Causes of Removal of ESM Packets V2.5 => V4.....	5
Table 5: Replaced ESM Packets V2.5 => V4.....	7
Table 6: Replaced DPM Packets V2.5 => V4.....	12
Table 7: Removed CoE Packets V2.5 => V4 .....	14
Table 8: Replaced CoE Packets V2.5 => V4.....	14
Table 9: Removed OD Packets V2.5 => V4.....	16
Table 10: Replaced OD Packets V2.5 => V4 .....	17
Table 11: Relationship between Parameter usListType of packet ECAT_OD_SDOINFO_GET_LIST_IND and Parameter usObjAccessMask of packet ODV3_GET_OBJECT_LIST_IND .....	23
Table 12: Allowed values for Parameter usAccessRights .....	25
Table 13: Error/diagnosis messages .....	27

## 5.2 Contacts

### Headquarters

#### Germany

Hilscher Gesellschaft für  
Systemautomation mbH  
Rheinstrasse 15  
65795 Hattersheim  
Phone: +49 (0) 6190 9907-0  
Fax: +49 (0) 6190 9907-50  
E-Mail: [info@hilscher.com](mailto:info@hilscher.com)

#### Support

Phone: +49 (0) 6190 9907-99  
E-Mail: [de.support@hilscher.com](mailto:de.support@hilscher.com)

### Subsidiaries

#### China

Hilscher Systemautomation (Shanghai) Co. Ltd.  
200010 Shanghai  
Phone: +86 (0) 21-6355-5161  
E-Mail: [info@hilscher.cn](mailto:info@hilscher.cn)

#### Support

Phone: +86 (0) 21-6355-5161  
E-Mail: [cn.support@hilscher.com](mailto:cn.support@hilscher.com)

#### France

Hilscher France S.a.r.l.  
69500 Bron  
Phone: +33 (0) 4 72 37 98 40  
E-Mail: [info@hilscher.fr](mailto:info@hilscher.fr)

#### Support

Phone: +33 (0) 4 72 37 98 40  
E-Mail: [fr.support@hilscher.com](mailto:fr.support@hilscher.com)

#### India

Hilscher India Pvt. Ltd.  
New Delhi - 110 065  
Phone: +91 11 26915430  
E-Mail: [info@hilscher.in](mailto:info@hilscher.in)

#### Italy

Hilscher Italia S.r.l.  
20090 Vimodrone (MI)  
Phone: +39 02 25007068  
E-Mail: [info@hilscher.it](mailto:info@hilscher.it)

#### Support

Phone: +39 02 25007068  
E-Mail: [it.support@hilscher.com](mailto:it.support@hilscher.com)

#### Japan

Hilscher Japan KK  
Tokyo, 160-0022  
Phone: +81 (0) 3-5362-0521  
E-Mail: [info@hilscher.jp](mailto:info@hilscher.jp)

#### Support

Phone: +81 (0) 3-5362-0521  
E-Mail: [jp.support@hilscher.com](mailto:jp.support@hilscher.com)

#### Korea

Hilscher Korea Inc.  
Seongnam, Gyeonggi, 463-400  
Phone: +82 (0) 31-789-3715  
E-Mail: [info@hilscher.kr](mailto:info@hilscher.kr)

#### Switzerland

Hilscher Swiss GmbH  
4500 Solothurn  
Phone: +41 (0) 32 623 6633  
E-Mail: [info@hilscher.ch](mailto:info@hilscher.ch)

#### Support

Phone: +49 (0) 6190 9907-99  
E-Mail: [ch.support@hilscher.com](mailto:ch.support@hilscher.com)

#### USA

Hilscher North America, Inc.  
Lisle, IL 60532  
Phone: +1 630-505-5301  
E-Mail: [info@hilscher.us](mailto:info@hilscher.us)

#### Support

Phone: +1 630-505-5301  
E-Mail: [us.support@hilscher.com](mailto:us.support@hilscher.com)