CC-Link
CC-Link IE

Control & Communication System Profile Specification

CSP+ Ver.2.2
BAP-C2008ENG-001-F

Revisions

Sub number	Revision description	Issued
*	The document number of this specification changed in line with the new numbering system for specifications (Previous document number: BAP-C1607-001)	February, 2013
A	Added specifications for CC-Link IE Safety Communication Function Support Unit Added specifications for Ethernet Connection Unit 1. Corrected CSP+ Specification Version to 2.0 4.3.1. Clarified the fact that items DEFAULT, RANGE, MIN_INC, ENG_UNIT, REF are not permitted with structural type elements 4.3.1.1. Clarified the fact that ACCESS items are not permitted in elements of STRUCT part 4.3.1.5. Clarified the fact that decimal and hexadecimal numbers can be written in the CODE item 4.3.1.8. Same as No. 2 4.3.1.17. Added SEQ_TARGET to keywords 4.3.1.17. Label names starting with "P_" are reserved because they are descriptions common to units 4.3.1.24. Added booleans to data types of elements that can be referenced to ENUM parts 4.3.1.25. Appended conditions when writing reference in the READ_DATA item 4.3.1.29. Written keyword "SEQ" in RELATED_MESSAGE, and added specifications for running multiple messages consecutively 4.3.1.30. Written keyword "SEQ" in RELATED_METHOD, and added specifications for running multiple methods consecutively 4.3.1.32. Clarified processing of empty bits when using STRUCT to write values in REQUEST_DATA (write 0) 4.3.1.38. Added conventions for TARGET items when using "SEQ" in RELATED_METHOD, RELATED_MESSAGE 4.3.1.40. Appended conditions when writing a reference in the WRITE_DATA item 4.3.1.40. Clarified processing of empty bits when using STRUCT to write values in WRITE_DATA (write 0) 4.3.3. Appended handling of control characters as an item common convention 5.2.1.1. (2) Addition concerning data types of Version Elements 5.2.1.1. (2) (e) Appended handling (priority) when the value of VersionDisplay elements change due to the version of CSP+ 5.2.1.1. (2) (f) Appended the fact that it is recommended not to change the value of VersionPolicy elements according to CSP+ version 5.2.1.3. (3) Appended the fact that when the value of DeviceModel elements are changed at the time CSP+ is updated, it is handled as a different model 5.2.1.3. (4) Appended the fact that the same ProductID should be written for the same model 5.5.1. Added the WarningStatus element to elements relating to the system area 5.5.3. Appended notes when using "SEQ" in a message, in the description of parts and elements concerning error history information acquisition	April, 2015

B	Added specifications for CC-Link IE Field Network Basic connection unit Added description specifications for acquiring energy management information Added description specifications for acquiring the status code and displaying detailed information of Ethernet connection unit 1. Corrected CSP+ Specification Version to 2.1. Defined a unit that has the SLMP (TCP/IP) connection function or CC-Link IE Field Network Basic connection function as an "Ethernet connection unit" 4.2.7. Added COMM_IF_INPUT, COMM_IF_OUTPUT, BLOCK_INPUT, BLOCK_OUTPUT, COMM_IF_COMMAND, and BLOCK_COMMAND to the CSP+ file structure of Ethernet connection unit 5.3.1.1. Deleted SyncMinTime element and CommnunicationMinTime element 5.3.1.3. Deleted the item descriptions of (63) SyncMinTime element and (64) CommnunicationMinTime element 5.5.8. Added conventions for acquiring energy management information 5.5.9. Added functions for acquiring status codes	January, 2017
C	Added description specifications for acquiring maintenance information 4.3.1.21. Added the type value "MAINTENANCE" and the detailed operating specifications of "MONITOR" 4.3.1.22. Added the type value "MAINTENANCE" and the detailed operating specifications of "MONITOR" 4.3.1.31. Added the keyword "POST" 4.3.1.32. Added the keyword "POST"	July, 2017
D	Added specifications for CC-Link IE TSN connection unit 1. Corrected CSP+ Specification Version to 3.0 4.1.1. Added arbitrary string to the CSP+ file name 4.1.2. Added the object dictionary 4.2.8. Added the CSP+ file structure of the CC-Link IE TSN connection unit 4.2.9. Added the information for the CC-Link IE TSN safety communication function support unit 4.3.1.3. Added addresses used in CC-Link IE TSN to ASSIGN conventions 4.3.1.6. Added the explanation when multiple comments are written to COMMENT conventions 4.3.1.9. Added the explanation for using "< >" for CC-Link IE TSN to DATA conventions 4.3.1.21. Added the message type that indicates the parameter automatic setting of the slave station to MESSAGE_TYPE 4.3.1.31. Added the explanation of RELATED_ELE conventions 5.2.1. Added elements related to CC-Link IE TSN support to DEVICE_INFO parts 5.3.1. Added elements related to CC-Link IE TSN support to COMM_IF_INFO parts 5.5.1. Added elements of CC-Link IE TSN to elements related to the system area 5.5.10.1. Added conventions related to error information 5.5.10.2. Added conventions related to MESSAGE for slave station parameter automatic setting 5.5.10.3. Added conventions related to parts and elements that indicate parameters for safety communication function 5.5.10.4. Added conventions related to MESSAGE specifications for writing parameters for safety communication function	November, 2018

E	Added specifications for CC-Link IE TSN connection unit 4.2.8.2. Added the note for setting multiple COMM_IF sections 4.3.1.37. Added description specifications of MESSAGE other than SLMP to REQUEST_TYPE conventions 5.3.1.1. Added AutoSettingHeaderType element 5.3.1.1. Deleted SyncMinTime element and CommnunicationMinTime element 5.3.1.3. (53) Added the note for writing multiple COMM_IF sections to the item description of StationMode element 5.3.1.3. (56) Added the item description of AutoSettingHeaderType element 5.3.1.3. Deleted the item descriptions of (63) SyncMinTime element and (64) CommnunicationMinTime element 5.3.7.1. Added the note for CC-Link IE TSN to the item of REQUEST_TYPE 5.5.10.1.1 Added description specifications for a profile which has multiple COMM_IF sections 5.5.10.2.2 Added the description specification when DATATYPE is constant to the MESSAGE part convention	April, 2019
F	1. Corrected CSP+ Specification Version to 3.1 4.1.1 Corrected the CSP+ file name and supplementary explanation 4.3.1.19 Deleted and added the reserved words (Deleted: RESPONSE_AREA, RESULT, and SELECT, Added: RESPONSE_DATA) in Table 4-13 5.3.1.1 Added CommunicationCycleMinTime_ns element and CommunicationCycleMinTime_s element to Figure 5-13 5.3.1.1 Added CommunicationCycleMinTime_ns element and CommunicationCycleMinTime_s element to Table 5-69 5.3.1.1 Added the descriptions that "CommunicationCycleMinTime_ns" may be omitted for devices whose minimum communication cycle is represented in the unit of second to *3 in Table 5-69. 5.3.1.3 (32) Added the value to No.6 in Table 5-106 5.3.1.3 (63) Added the item description of CommunicationCycleMinTime_ns 5.3.1.3 (64) Added the item description of CommunicationCycleMinTime_s 5.5.6.2 (2) (a) Corrected all the figures in Figure 5-49 5.5.10.1.1 Added the specification to be referred to for the error details information 5.5.10.4.1 Corrected supplementary explanation for Table 5-444 5.5.10.4.1 Added supplementary explanation for Table 5-445 5.5.10.4.3 (a) Corrected xml	November, 2019

Table of Contents

1. FOREWORD 15

2. OVERVIEW 16
2.1 What is CSP+? 16
2.2 FEATURES 17
2.3 Flow of CSP+ File Utilization 18

3. GENERAL CONCEPT OF CSP+ FILE DESCRIPTION SPECIFICATIONS 19
3.1 Model of CC-Link Family Connected Module 19
3.2 Models by Module Access Method 20
3.2.1 Access by cyclic transmission 20
3.2.2 Access by transient transmission 23
3.3 Models by Module Form 26

4. CSP+ FILE STRUCTURE AND DESCRIPTION SPECIFICATIONS 30
4.1 CSP+ File Management Specifications 30
4.1.1 CSP+ file name 30
4.1.2 Data configuration with compression management 31
4.2 CSP+ File Structure 32
4.2.1 Definitions of sections within a CSP+ file 32
4.2.2 Definitions of parts within a CSP+ file section 33
4.2.3 Definitions of elements and items of CSP+ parts 36
4.2.4 CSP+ file structure of CC-Link connected module 38
4.2.5 CSP+ file structure of CC-Link IE Field Network connected modules 43
4.2.6 In the case of the CC-Link IE Field Network Safety Communication Function Support Unit 48
4.2.7 CSP+ file structure of Ethernet connection unit 49
4.2.8 CSP+ file structure of the CC-Link IE TSN connection unit 52
4.2.9 In the case of the CC-Link IE TSN safety communication function support unit 54
4.2.10 Structure of sections, parts, and elements of a CSP+ file 55
4.3 SCP+ File Description Specifications 57
4.3.1 Item conventions 57
4.3.2 Prohibited XML character support 84
4.3.3 Common conventions related to item values 85
4.3.4 Conventions for references that use labels 90

5. CSP+ File Details 94
5.1 FILE Section 95
5.1.1 FILE_INFO part 96
5.2 DEVICE Section 104
5.2.1 DEVICE_INFO part 105
5.3 COMM_IF Section 133
5.3.1 COMM_IF_INFO part 134
5.3.2 COMM_IF_INPUT part 210
5.3.3 COMM_IF_OUTPUT part 215
5.3.4 COMM_IF_PARAMETER part 218
5.3.5 COMM_IF_COMMAND part 223
5.3.6 METHOD part 227
5.3.7 MESSAGE part 235
5.3.8 STRUCT part 242
5.3.9 ENUM part 248
5.3.10 COMMAND_ARGUMENT part 254
5.4 BLOCK Section 259
5.4.1 BLOCK_INFO part 260
5.4.2 BLOCK_INPUT part 265
5.4.3 BLOCK_OUTPUT part 270
5.4.4 BLOCK_PARAMETER part 273
5.4.5 BLOCK_COMMAND part 278
5.4.6 STRUCT part 281
5.4.7 ENUM part 283
5.4.8 COMMAND_ARGUMENT part 285
5.4.9 CONDITION Parts 286
5.5 Parts and Elements with Defined Applications 294

5.5.1 Elements related to system area 295
5.5.2 Parts and elements related to parameter initialization based on initial settings 321
5.5.3 Parts and elements related to acquisition of error logs 324
5.5.4 Method Specifications for conducting error checks of equipment 340
5.5.5 Parts and elements that indicated parameters for safety communication function 345
5.5.6 MESSAGE Specifications for parameter writing of safety communication functions 348
5.5.7 Conventions of read/write capable parameters using parameter IDs 353
5.5.8 Parts and elements related to the acquisition of energy management information 356
5.5.9 Convention for acquiring status codes 387
5.5.10 Specifications related to CC-Link IE TSN 392

6. CSP+ Description Method 418
6.1 CSP+ Version Management Method 418
6.2 Other Language Management 419
6.3 CSP+ File Management 420

7. Control & Communication System Profile XML Scheme 421
7.1 XML Schema 421

8. Description Specifications of Object Dictionary File 432
8.1 CSV File Format 432
8.2 Description Specifications for Each Item 433

9. Additional specifications for CSP+ Specifications Version 1.1 or later 435
9.1 Additional specifications for CSP+ Specifications Version 1.1 435
9.2 Additional specifications for CSP+ Specifications Version 2.0 435
9.3 Additional specifications for CSP+ specifications Version 2.1 436
9.4 Additional specifications for CSP+ specifications Version 2.2 436
9.5 Additional Specifications for CSP+ Specification Version 3.0 437

10. Related Specification Documents 438

Figure 2-1 CC-Link Family System Profile (CSP+) 18
Figure 3-1 Model of CSP+ Module 19
Figure 3-2 Example of Input/Output by Cyclic Transmission 20
Figure 3-3 Example of Parameter Transmission/Reception by Cyclic Transmission 21
Figure 3-4 Example of Command Transmission by Cyclic Transmission 22
Figure 3-5 Input/Output by SLMP 23
Figure 3-6 Parameter Transmission/Reception by SLMP 24
Figure 3-7 Command Transmission by SLMP 25
Figure 3-8 Model of Module Connected to Single Network 26
Figure 3-9 Model of Module Used While Switching Between Multiple Networks 26
Figure 3-10 Model of Module When Only Communication Head Module Is Described 27
Figure 3-11 Model of Module When CSP+ File Is Written Based on a Configuration that Includes Connected Modules (When Virtual Control Devices Are Separately Described for Communication Head Module and Other Modules) 28
Figure 3-12 Model of a Module When a CSP+ File Is Written Based on a Configuration that Includes Connected Modules (When the Virtual Control Devices of the Communication Head Module and Other Modules Are Described Together) 29
Figure 4-1 Configuration within Compressed File 31
Figure 4-2 CSP+ File Structure 32
Figure 4-3 Section Structure 33
Figure 4-4 Part Structure 36
Figure 4-5 General Model of CSP+ File for CC-Link Connected Module 39
Figure 4-6 Model of CSP+ File for CC-Link Remote I/O Station 40
Figure 4-7 Model of CSP+ File for CC-Link Remote Device Station, Intelligent Device Station, and Local Station 41
Figure 4-8 Description Example of Multiple COMM_IF Sections with CC-Link 42
Figure 4-9 General Model of CSP+ File of CC-Link IF Field Network Connected Module 44

Figure 4-10 Model of CSP+ File for CC-Link IE Field Network Connected Remote I/O 45
Figure 4-11 Model of CSP+ File of CC-Link IE Field Network Connected Remote Device Station, Intelligent Device Station, and Local Station 46
Figure 4-12 Writing Multiple COMM_IF Sections in a CC-Link IE Field Network 47
Figure 4-13 General Model of CSP+ File for Ethernet Connection Unit 50
Figure 4-14 Description Example of Multiple COMM_IF Sections with Ethernet Connection Unit 51
Figure 4-15 General Model of CSP+ File of CC-Link IE TSN Connection Unit 52
Figure 4-16 Example of CC-Link IE TSN Devices for which Station Mode Can Be Changed 53
Figure 4-17 Structure of FILE Section 55
Figure 4-18 Structure of DEVICE Section 55
Figure 4-19 Structure of COMM_IF Section 56
Figure 4-20 Structure of BLOCK Section 56
Figure 5-1 Structure of FILE Section 95
Figure 5-2 Structure of FILE_INFO Part 96
Figure 5-3 Structure of Element of FILE_INFO Part 96
Figure 5-4 Structure of DEVICE Section 104
Figure 5-5 Structure of DEVICE_INFO Part 105
Figure 5-6 Structure of DEVICE_INFO Part for tool link in CC-Link IE TSN 105
Figure 5-7 Structure of Element of DEVICE_INFO Part 106
Figure 5-8 Device Version Management Example 110
Figure 5-9 Structure of COMM_IF Section 133
Figure 5-10 Structure of COMM_IF_INFO parts (in the case of CC-Link) 134
Figure 5-11 Structure of COMM_IF_INFO Part (in the case of CC-Link IE Field Network) 135
Figure 5-12 Structure of COMM_IF_INFO Parts (in the case of Ethernet) 136
Figure 5-13 Structure of COMM_IF_INFO Part (CC-Link IE TSN) 137
Figure 5-14 Detailed Structure of When a Safety Communication Function of CC-Link IE TSN is Available 138
Figure 5-15 Structure Definition for Element of COMM_IF_INFO Part 138
Figure 5-16 Structure Definition of Element of COMM_IF_INPUT and COMM_IF_OUTPUT Part 210
Figure 5-17 Structure Definition for Element of COMM_IF_PARAMETER Part 218
Figure 5-18 Structure Definition for Element of COMM_IF_COMMAND Part 223
Figure 5-19 Structure Definition for Element of METHOD Part 227
Figure 5-20 Flow of Procedure-Based Communication Service 231
Figure 5-21 Operation of Communication Service Described in METHOD 232
Figure 5-22 METHOD Sequence 233
Figure 5-23 Structure Definition for Element of MESSAGE Part 235
Figure 5-24 Operation of Communication Service Described in MESSAGE 239
Figure 5-25 MESSAGE Call-up and Operation Sequence 240
Figure 5-26 Structure Definition for Element of STRUCT Part 242
Figure 5-27 Structure Definition for Element of ENUM Part 248
Figure 5-28 Structure Definition for Element of COMMAND_ARGUMENT Part 254
Figure 5-29 Structure of BLOCK Section 259
Figure 5-30 Structure of BLOCK_INFO Part 260
Figure 5-31 Structure Definition for Element of BLOCK_INFO Part 260
Figure 5-32 Structure Definition for Element of BLOCK_INPUT Part and BLOCK_OUTPUT Part 265
Figure 5-33 Structure Definition for Element of BLOCK_PARAMETER Part 273
Figure 5-34 Structure Definition for Element of BLOCK_COMMAND Part 278
Figure 5-35 Structure of CONDITION Parts 286
Figure 5-36 Structure of conditionVariable parts in CONDITION Parts 286

Figure 5-37 Structural definition of conditionMemmber parts in CONDITION Parts 287
Figure 5-38 Example of conditional expression (CONDITION) 292
Figure 5-39 Relationship between conditional expressions (CONDITION) and parameters 293
Figure 5-40 Image of Structure When System Area Is Described 296
Figure 5-41 Image of Structure When Parameter Initialization Based on Initial Settings Is Described 321
Figure 5-42 Image of Structure When Error Log Acquisition Is Described (When Error Logs Are Managed on Module Main Body Side) 324
Figure 5-43 Image of Structure When Error Log Acquisition Is Described (When Error Logs Are Managed on Communication Module Side) 325
Figure 5-44 Structural image when conducting error check of the equipment 340
Figure 5-45 Examples of METHOD & COMMAND that perform error check of equipment (some unwritable or optional items omitted) 344
Figure 5-46 Structural image in when writing parameters of safety communication functions 345
Figure 5-47 Structural image when describing parameter writing of safety communication functions 348
Figure 5-48 Structural image when describing enabling parameters of safety communication functions 348
Figure 5-49 Example of MESSAGE for parameter writing of safety communication functions 352
Figure 5-50 Structural image of read/write capable parameters using parameter IDs 353
Figure 5-51 Image of Structure When Acquisition of Energy Management Information Is Described (When Energy Management Information Is Managed on Module Main Body Side) 356
Figure 5-52 Image of Structure When Acquisition of Energy Management Information Is Described (When Energy Management Information Is Managed on Communication Module Side) 356
Figure 5-53 Image of Structure When Acquisition of Status Information Is Described 387
Figure 5-54 Image of Structure When Describing Error Related Information Acquisition 392
Figure 5-55 Description Example of Parts and Elements Related to Error Related Information 400
Figure 5-56 Image of Structure When Describing MESSAGE for Slave Station Parameter Automatic Setting 401
Figure 5-57 Image of Structure When Describing Safety Communication Function Parameters 406
Figure 5-58 Image of Structure When Describing Parameter Writing of Safety Communication Function 411
Figure 5-59 Image of Structure When Describing Enabling of Safety Communication Function Parameters 411
Figure 6-1 Version Update Example 419
Figure 8-1 CSV File Output Example 432

Table 4-1 Parts of a FILE Section 33
Table 4-2 Parts of a DEVICE Section 33
Table 4-3 Parts of a COMM_IF Section 34
Table 4-4 Parts of a BLOCK Section 35
Table 4-5 Items of Elements of Parts 36
Table 4-6 Items of Elements of METHOD and MESSAGE Parts 36
Table 4-7 Item List 37
Table 4-8 Definitions of Modules by I/O Type 40
Table 4-9 ACCESS Attribute List 58
Table 4-10 Data Type List 63
Table 4-11 Error PDU Name List 66
Table 4-12 Applicable Relational Operators 67
Table 4-13 Characters Not Permitted in LABEL and LABEL2 Items 68

Table 4-14 Values of MESSAGE_TYPE Item 69
Table 4-15 Values of METHOD_TYPE Item 70
Table 4-16 RELATED_MESSAGE Item Conventions 76
Table 4-17 RELATED_METHOD Item Conventions 78
Table 4-18 PDU Name List for SLMP Request 79
Table 4-19 PDU Name List for SLMP Response 80
Table 4-20 Items that Cannot Be Written in Elements WhereSEQ_TARGET Is Written 81
Table 4-21 List of Input Items of Common Utility Software Display Information 82
Table 4-22 Entity References for Prohibited Characters 84
Table 4-23 Description Rules 85
Table 4-24 List of Conventions Applicable to Each Item 87
Table 4-25 Convention Combined Use Rules 89
Table 4-26 List of Reference Writable Items 90
Table 4-27 Description Method of Data Size of Element 93
Table 5-1 CSP+ XML Element List 94
Table 5-2 profile Element Attribute List 94
Table 5-3 List of XML Child Elements of profile Element 95
Table 5-4 List of Attributes of file Element 95
Table 5-5 List of XML Child Elements of file Element 95
Table 5-6 Element List for FILE_INFO Part 97
Table 5-7 Item List for FILE_INFO Part 97
Table 5-8 Attribute List for fileInfo Element 98
Table 5-9 XML Child Element List for fileInfo Element 98
Table 5-10 Attribute List for fileInfoMember Element 98
Table 5-11 XML Child Element List of fileInfoMember Element 98
Table 5-12 Item Description Specifications of CreateDate Element 99
Table 5-13 Item Description Specifications of CreateTime Element 99
Table 5-14 Item Description Specifications of ModDate Element 100
Table 5-15 Item Description Specifications of ModTime Element 100
Table 5-16 Item Description Specifications of Language Element 101
Table 5-17 Item Description Specifications of CCLinkFamilyProfileVersionElement 102
Table 5-18 Item Description Specifications of FileVersionElement 103
Table 5-19 File Version Upgrade Rules 103
Table 5-20 List of Attributes of device Element 104
Table 5-21 List of XML Child Elements of device Element 104
Table 5-22 Element List for DEVICE_INFO Part 107
Table 5-23 Additional Element of DEVICE_INFO Part in CC-Link IE TSN 108
Table 5-24 Item List for DEVICE_INFO Part 109
Table 5-25 Meaning of Each Value of VersionPolicy and Example of Modules that Use the Value 111
Table 5-26 Attribute List for deviceInfo Element 113
Table 5-27 XML Child Element List for deviceInfo Element 113
Table 5-28 Attribute List for deviceInfoMember Element 113
Table 5-29 XML Child Element List of deviceInfoMember Element 113
Table 5-30 Item Description Specifications of VendorName Element 114
Table 5-31 Item Description Specifications of VendorCode Element 114
Table 5-32 Item Description Specifications of DeviceModel Element 115
Table 5-33 Item Description Specifications of ProductID Element 116
Table 5-34 Item Description Specifications of DeviceTypeID Element 116
Table 5-35 Item Description Specifications of DeviceTypeDetail Element 117
Table 5-36 Item Description Specifications of Version Element 118

Table 5-37 Item Description Specifications of VersionDisplayFlg Element 119
Table 5-38 Item Description Specifications of VersionPolicyType Element 119
Table 5-39 Item Description Specifications of DisplayVersionValue Element 120
Table 5-40 Item Description Specifications of VersionComment Element 120
Table 5-41 Item Description Specifications of DeviceConfigurationID Element 121
Table 5-42 Item Description Specifications of DeviceConfigurationComment Element 121
Table 5-43 Item Description Specifications of ReferenceURL Element 122
Table 5-44 Item Description Specifications of URLInfo Element 122
Table 5-45 Item Description Specifications of Outline Element 123
Table 5-46 Item Description Specifications of Feature Element 123
Table 5-47 Item Description Specifications of SpecList Element 124
Table 5-48 Item Description Specifications of PowerSupplyVoltage Element 124
Table 5-49 Item Description Specifications of ConsumptionCurrent Element 125
Table 5-50 Item Description Specifications of IconFileName Element 125
Table 5-51 Item Description Specifications of GraphicsFileName Element 126
Table 5-52 Item Description Specifications of Height Element 126
Table 5-53 Item Description Specifications of Width Element 127
Table 5-54 Item Description Specifications of Depth Element 127
Table 5-55 Item Description Specifications of Weight Element 128
Table 5-56 Item Description Specifications of Price Element 128
Table 5-57 Item Description Specifications of UI_ATTRIBUTE_Window** Element 129
Table 5-58 Item Description Specifications of DedicatedToolFlg Element 130
Table 5-59 Item Description Specifications of DedicatedToolName Element 130
Table 5-60 Item Description Specifications of InstallRegistryKeyName Element 131
Table 5-61 Item Description Specifications of InstallRegistryValueName Element 131
Table 5-62 Item Description Specifications of ExePathRegistryKeyName Element 132
Table 5-63 Item Description Specifications of ExePathRegistryValueName Element 132
Table 5-64 List of Attributes of commIf Element 133
Table 5-65 List of XML Child Elements of commIf Element 133
Table 5-66 Elements of COMM_IF_INFO Part with CC-Link 139
Table 5-67 Elements of COMM_IF_INFO Part with CC-Link IE Field Network 140
Table 5-68 Elements of COMM_IF_INFO parts in Ethernet 141
Table 5-69 Elements of COMM_IF_INFO Part with CC-Link IE TSN 142
Table 5-70 Item List for COMM_IF_INFO Part 144
Table 5-71 Attribute List for commIfInfo Element 148
Table 5-72 XML Child Element List for commIfInfo Element 148
Table 5-73 Attribute List for commIfInfoMember Element 148
Table 5-74 XML Child Element List of commIfInfoMember Element7 149
Table 5-75 Item Description Specifications of VendorName Element 149
Table 5-76 Item Description Specifications of VendorCode Element 150
Table 5-77 Item Description Specifications of CommIFTypeID Element 150
Table 5-78 Item Description Specifications of Version Element 151
Table 5-79 Item Description Specifications of ReadVersionType Element 152
Table 5-80 Item Description Specifications of StationType Element 153
Table 5-81 Item Description Specifications of StationTypeDetail Element 153
Table 5-82 Item Description Specifications of TotalNumOfIO Element 154
Table 5-83 Item Description Specifications of IOType Element 154
Table 5-84 Item Description Specifications of DevModel Element 155
Table 5-85 Item Description Specifications of CcLinkVer Element 155
Table 5-86 Item Description Specifications of ExtCycle Element 156
Table 5-87 Item Description Specifications of NumOccupiedStations Element 156

Table 5-88 Item Description Specifications of ErrReg Element 157
Table 5-89 Item Description Specifications of MasterFlg Element 157
Table 5-90 Item Description Specifications of VendorName2 Element 158
Table 5-91 Item Description Specifications of nodeType Element 158
Table 5-92 Item Description Specifications of IOType Element 159
Table 5-93 Item Description Specifications of ModelCode Element 159
Table 5-94 Item Description Specifications of DevModel Element 160
Table 5-95 Item Description Specifications of ModelName Element 160
Table 5-96 Item Description Specifications of RYSize Element 161
Table 5-97 Item Description Specifications of RWwSize Element 161
Table 5-98 Item Description Specifications of RXSize Element 162
Table 5-99 Item Description Specifications of RWrSize Element 162
Table 5-100 Item Description Specifications of Ports Element 163
Table 5-101 Item Description Specifications of protcolVersion Element 163
Table 5-102 Item Description Specifications of NodeNumberSettingFig Element 164
Table 5-103 Item Description Specifications of TransientReceptionFig Element 164
Table 5-104 Item Description Specifications of SMLPReceptionFig Element 165
Table 5-105 Item description specifications of SafetyProtocolVersion element 165
Table 5-106 Item description specifications of SafetyNodeType element 166
Table 5-107 Item description specifications of SafetyIn_XSize element 166
Table 5-108 Item description specifications of SafetyOut_YSize element 167
Table 5-109 Item description specifications of SupportSafetyCommService element 167
Table 5-110 Services shown by each bit 167
Table 5-111 Item description specifications of SupportIPAddressSize 168
Table 5-112 Item description specifications of TimeOutValue element 168
Table 5-113 Item description specifications of SupportFlg_SearchNode 169
Table 5-114 Item description specifications of SupportFlg_SetIPAddress element 169
Table 5-115 Item description specifications of SupportFlg_CompareDeviceInfo 170
Table 5-116 Item description specifications of SupportFlg_GetParam element 170
Table 5-117 Item description specifications of SupportFlg_SetParam 171
Table 5-118 Item description specifications of SupportFlg_StartSetParam element 171
Table 5-119 Item description specifications of SupportFlg_EndSetParam 172
Table 5-120 Item description specifications of SupportFlg_CancelSetParam element 172
Table 5-121 Item description specifications of SupportFlg_ReadStatus 173
Table 5-122 Item description specifications of SupportFlg_GetCommSetting element 173
Table 5-123 Item description specifications of SupportFlg_ReadStatus2 element 174
Table 5-124 Item Description Specifications of DevModel Element 174
Table 5-125 Item Description Specifications of NumOccupiedStations Element 175
Table 5-126 Item Description Specifications of CCIEFBasicProtocolVersion Element 175
Table 5-127 Item Description Specifications of EthernetCommFunction Element 176
Table 5-128 Item Description Specifications of StationMode Element 177
Table 5-129 Item Description Specifications of StationModeName Element 177
Table 5-130 Item Description Specifications of LocalFunction Element 178
Table 5-131 Item Description Specifications of AutoSettingHeaderType Element 178
Table 5-132 Item Description Specifications of AutoSettingType Element 179
Table 5-133 Item Description Specifications of ModelCode Element 180
Table 5-134 Item Description Specifications of ModelCodeEx Element 180
Table 5-135 Item Description Specifications of DevModel Element 181
Table 5-136 Item Description Specifications of CanProfileNum Element 181
Table 5-137 Item Description Specifications of ObjectDictionaryFileName Element 182
Table 5-138 Item Description Specifications of CommunicationCycleMinTime_ns Element 182

Table 5-139 Item Description specifications of CommunicationCycleMinTime_s Element 183
Table 5-140 Item Description Specifications of IEEE802_1ASFunction Element 183
Table 5-141 Item Description Specifications of ReceiveFunction100M Element 184
Table 5-142 Item Description Specifications of RelayFunction100M Element 184
Table 5-143 Item Description Specifications of ReceiveFunction1G Element 185
Table 5-144 Item Description Specifications of RelayFunction1G Element 185
Table 5-145 Item Description Specifications of MultiCastFunction Element..... 186
Table 5-146 Item Description Specifications of CertificationClass Element 186
Table 5-147 Item Description Specifications of S_B_DefaultSize Element 187
Table 5-148 Item Description Specifications of S_W_DefaultSize Element 187
Table 5-149 Item Description Specifications of R_B_DefaultSize Element 188
Table 5-150 Item Description Specifications of R_W_DefaultSize Element 188
Table 5-151 Item Description Specifications of L_B_DefaultSize Element..... 189
Table 5-152 Item Description Specifications of L_W_DefaultSize Element..... 189
Table 5-153 Item Description Specifications of S_B_MaxSize Element 190
Table 5-154 Item Description Specifications of S_W_MaxSize Element 190
Table 5-155 Item Description Specifications of R_B_MaxSize Element..... 191
Table 5-156 Item Description Specifications of R_W_MaxSize Element 191
Table 5-157 Item Description Specifications of L_B_MaxSize Element 192
Table 5-158 Item Description Specifications of L_W_MaxSize Element 192
Table 5-159 Item Description Specifications of S_B_MinSize Element 193
Table 5-160 Item Description Specifications of S_W_MinSize Element 193
Table 5-161 Item Description Specifications of R_B_MinSize Element..... 194
Table 5-162 Item Description Specifications of R_W_MinSize Element..... 194
Table 5-163 Item Description Specifications of L_B_MinSize Element 195
Table 5-164 Item Description Specifications of L_W_MinSize Element 195
Table 5-165 Item Description Specifications of S_B_Address Element 196
Table 5-166 Item Description Specifications of S_W_Address Element 196
Table 5-167 Item Description Specifications of R_B_Address Element 197
Table 5-168 Item Description Specifications of R_W_Address Element 197
Table 5-169 Item Description Specifications of StsW_Address Element..... 198
Table 5-170 Item Description Specifications of PDOConfigIndex+X Element 198
Table 5-171 Item Description Specifications of PDOConfigPDOType+X Element 199
Table 5-172 Item Description Specifications of PDOConfigMemoryAddress+X Element..... 199
Table 5-173 Item Description Specifications of PDOConfigPossibleMapping+X Element 200
Table 5-174 Item Description Specifications of S_General_Address Element 200
Table 5-175 Item Description Specifications of R_General_Address Element 201
Table 5-176 Item Description Specifications of SafetyProtocolVersion Element 202
Table 5-177 Item Description Specifications of SafetyNodeType Element..... 203
Table 5-178 Item Description Specifications of SupportSafetyCommService Element 204
Table 5-179 Service Shown by Each Bit 204
Table 5-180 Item Description Specifications of SupportSafetyInvokeFunc Element..... 205
Table 5-181 Safety Function Shown by Each Bit..... 205
Table 5-182 Item Description Specifications of S_Safe_DefaultSize Element 206
Table 5-183 Item Description Specifications of S_Safe_MaxSize Element 206
Table 5-184 Item Description Specifications of S_Safe_MinSize Element 207
Table 5-185 Item Description Specifications of S_Safe_Address Element..... 207
Table 5-186 Item Description Specifications of R_Safe_DefaultSize Element 208
Table 5-187 Item Description Specifications of R_Safe_MaxSize Element 208
Table 5-188 Item Description Specifications of R_Safe_MinSize Element 209
Table 5-189 Item Description Specifications of R_Safe_Address Element 209

Table 5-190 List of Elements Defined in COMM_IF_Input Part and COMM_IF_OUTPUT Part 211
Table 5-191 Attribute List for commIfInput Element 212
Table 5-192 XML Child Element List for commIfInput Element 212
Table 5-193 Attribute List for commIfInputMember Element 212
Table 5-194 XML Child Element List of commIfInputMember Element 212
Table 5-195 Item Description Example of Element 214
Table 5-196 Attribute List for commIfOutput Element 215
Table 5-197 XML Child Element List for commIfOutput Element 215
Table 5-198 Attribute List for commIfOutputMember Element 215
Table 5-199 XML Child Element List of commIfOutputMember Element 215
Table 5-200 Item Description Example of Element 217
Table 5-201 Item List for Elements Defined in COMM_IF_PARAMETER Part 219
Table 5-202 Attribute List for commIfParameter Element 220
Table 5-203 XMLChild Element List for commIfParameterElement 220
Table 5-204 Attribute List for commIfParameterMember Element 220
Table 5-205 XML Child Element List of commIfParameterMember Element 220
Table 5-206 Item Description Example of Element 222
Table 5-207 Item List for Elements Defined in COMM_IF_COMMAND Part 223
Table 5-208 Attribute List for commIfCommand Element 224
Table 5-209 XML Child Element List for commIfCommandElement 224
Table 5-210 Attribute List for commIfPCommandMember Element 224
Table 5-211 XML Child Element List of commIfCommandMember Element 224
Table 5-212 Item Description Example of Element 226
Table 5-213 Item List for Elements Defined in METHOD Part 228
Table 5-214 Attribute List for method Element 229
Table 5-215 XML Child Element List for methodElement 229
Table 5-216 Attribute List for methodMember Element 229
Table 5-217 XML Child Element List of methodMember Element 229
Table 5-218 Item Description Example of Element 234
Table 5-219 Item List for Elements Defined in MESSAGE Part 236
Table 5-220 Attribute List for message Element 237
Table 5-221 XML Child Element List of message Element 237
Table 5-222 Attribute List for messageMember Element 237
Table 5-223 XML Child Element List of messageMember Element 237
Table 5-224 Item Description Example of Element 241
Table 5-225 Item List for Elements Defined in STRUCT Part 243
Table 5-226 Attribute List for struct Element 244
Table 5-227 XML Child Element List for struct Element 244
Table 5-228 Attribute List for structMember Element 244
Table 5-229 XML Child Element List of structMember Element 244
Table 5-230 Item Description Example of Element (ConvertEnable1) 246
Table 5-231 Item Description Example of Element (ConvertEnable2) 246
Table 5-232 Item Description Example of Element (ConvertEnable3) 246
Table 5-233 Item Description Example of Element (ConvertEnable4) 246
Table 5-234 Item List for Elements Defined in ENUM Part 249
Table 5-235 Attribute List for enum Element 250
Table 5-236 XML Child Element List for enumElement 250
Table 5-237 Attribute List for enumMember Element 250
Table 5-238 XML Child Element List of enumMember Element 250
Table 5-239 XML Child Element List of enumMember Element (For CC-Link IE TSN) 251
Table 5-240 Item Description Example of Element (SelCH1) 252

Table 5-241 Item Description Example of Element (SelCH2) 252
Table 5-242 Item Description Example of Element (SelCH3) 252
Table 5-243 Item Description Example of Element (SelCH4) 252
Table 5-244 Item Description Example of Element (enum2) 252
Table 5-245 Item List for Elements Defined in COMMAND_ARGUMENT Part 255
Table 5-246 Attribute List for commandArgument Element 256
Table 5-247 XML Child Element List for commandArgumentElement 256
Table 5-248 Attribute List for commandArgumentMember Element 256
Table 5-249 XML Child Element List of commandArgumentMember Element 256
Table 5-250 Item Description Example of Element 258
Table 5-251 List of Attributes of block Element 259
Table 5-252 List of XML Child Elements of block Element 259
Table 5-253 Elements of BLOCK Part 261
Table 5-254 Item List of BLOCK_INFO Part 261
Table 5-255 Attribute List for blockInfo Element 262
Table 5-256 XML Child Element List for blockInfo Element 262
Table 5-257 Attribute List for blockInfoMember Element 262
Table 5-258 XML Child Element List of blockIfInfoMember Element 262
Table 5-259 Item Description Specifications of VendorName Element 263
Table 5-260 Item Description Specifications of VendorCode Element 263
Table 5-261 Item Description Specifications of Version Element 264
Table 5-262 List of Items Defined inBLOCK_INPUT Part and BLOCK_OUTPUT Part 266
Table 5-263 Attribute List for blockInput Element 267
Table 5-264 XML Child Element List for blockInput Element 267
Table 5-265 Attribute List for blockInputMember Element 267
Table 5-266 XML Child Element List of blockIfInputMember Element 267
Table 5-267 Item Description Example of Element 269
Table 5-268 Attribute List for blockOutput Element 270
Table 5-269 XML Child Element List for blockOutput Element 270
Table 5-270 Attribute List for blockOutputMember Element 271
Table 5-271 XML Child Element List of blockOutputMember Element 271
Table 5-272 Item Description Example of Element 272
Table 5-273 Item List for Elements Defined in BLOCK_PARAMETER Part 274
Table 5-274 Attribute List for blockParameter Element 275
Table 5-275 Child Element List for blockParameter Element 275
Table 5-276 Attribute List for blockParameterMember Element 275
Table 5-277 Child Element List of blockParameterMember Element 275
Table 5-278 Item Description Example of Element 277
Table 5-279 Item List for Elements Defined in BLOCK_COMMAND Part 278
Table 5-280 Attribute List for blockCommand Element 279
Table 5-281 XML Child Element List for blockCommand Element 279
Table 5-282 Attribute List for blockCommandMember Element 279
Table 5-283 List of XML Child Elements of blockCommandMemberElement 279
Table 5-284 Item Description Example of Element 280
Table 5-285 Item Description Example of Element (STReturnCode1) 281
Table 5-286 Item Description Example of Element (STReturnCode2) 281
Table 5-287 Item Description Example of Element 283
Table 5-288 Item Description Example of Element 283
Table 5-289 Item Description Example of Element 283
Table 5-290 Item Description Example of Element 283
Table 5-291 Item Description Example of Element (enum2) 284

Table 5-292 Item Description Example of Element 285
Table 5-293 List of conditionVariable parts of CONDITION Parts 287
Table 5-294 List of conditionMember parts of CONDITION Parts 287
Table 5-295 List of condition element attributes 288
Table 5-296 List of XML subelements of the condition element 288
Table 5-297 List of XML subelements of conditionVariable parts 288
Table 5-298 List of XML subelements of conditionMember parts 288
Table 5-299 List of attributes of the conditionXVariable element 288
Table 5-300 List of attributes of the conditionXVariable element 289
Table 5-301 List of attributes of the conditionXVariable element 289
Table 5-302 Item description specifications of conditionVariable parts 290
Table 5-303 Item description specifications of conditionMemmber parts 291
Table 5-304 Network Usage List for Each Application 294
Table 5-305 Element List When the Communication Interface Described in the COMM_IF_INPUT Part is CC-Link 297
Table 5-306 Element List When the Communication Interface Described in the COMM_IF_INPUT Part is the CC-Link IE Field Network 297
Table 5-307 Element List When the Communication Interface Described in the COMM_IF_INPUT Part is CC-Link IE TSN 297
Table 5-308 Element List When the Communication Interface Described in the COMM_IF_OUTPUT Part is CC-Link 297
Table 5-309 Element List When the Communication Interface Described in the COMM_IF_OUTPUT Part is the CC-Link Field Network 297
Table 5-310 Element List When the Communication Interface Described in the COMM_IF_OUTPUT Part is CC-Link IE TSN 297
Table 5-311 Item Description Example of InitialDataProcessReq Element 298
Table 5-312 Item Description Specifications of InitialDataSetComp Element 299
Table 5-313 Item Description Specifications of ErrorStatus Element 300
Table 5-314 Item Description Specifications of RemoteReady Element 301
Table 5-315 Item Description Specifications of MessageTransReceive Element 302
Table 5-316 Item Description Specifications of MessageHandshakeRX Element 303
Table 5-317 Item Description Specifications of WarningStatus Element 304
Table 5-318 Item Description Example of InitialDataProcessingReq Element 305
Table 5-319 Item Description Specifications of InitialDataSetComp Element 306
Table 5-320 Item Description Specifications of ErrorStatus Element 307
Table 5-321 Item Description Specifications of RemoteReady Element 308
Table 5-322 Item Description Example of WarningStatus Element 309
Table 5-323 Item Description Specifications of ErrorStatus Element 310
Table 5-324 Item Description Specifications of RemoteReady Element 311
Table 5-325 Item Description Example of InitialDataProcessComp Element 312
Table 5-326 Item Description Specifications of InitialDataSetReq Element 313
Table 5-327 Item Description Specifications of ErrorResetReq Element 314
Table 5-328 Item Description Specifications of MessageTransReq Element 315
Table 5-329 Item Description Specifications of MessageHandshakeRY Element 316
Table 5-330 Item Description Example of InitialDataProcessComp Element 317
Table 5-331 Item Description Specifications of InitialDataSetReq Element 318
Table 5-332 Item Description Specifications of ErrorResetReq Element 319
Table 5-333 Item Description Specifications of ErrorResetReq Element 320
Table 5-334 List of Elements Described in METHOD Part 321
Table 5-335 List of Elements Described in COMM_IF_PARAMETER Part 321
Table 5-336 Item Description Specifications of ParamInitialize Element 322
Table 5-337 Item Description Specifications of Parameter1 Element 323

Table 5-338 List of Elements Described in MESSAGE Part 325
Table 5-339 List of Elements Described in COMM_IF_COMMAND Part 325
Table 5-340 List of Elements Described in BLOCK_COMMAND Part 325
Table 5-341 List of Elements Described in COMMAND_ARGUMENT Part 326
Table 5-342 Item Description Specifications for SLMPGetErrorLogMessage* Element 327
Table 5-343 Item Description of GetErrorLogCommand** Element 329
Table 5-344 Item Description Specifications of GetErrorLogCommand** Element 330
Table 5-345 Item Description Specifications of ErrorCode Element 331
Table 5-346 Item Description Specifications of ErrorYear Element 332
Table 5-347 Item Description Specifications of ErrorMonth Element 333
Table 5-348 Item Description Specifications of ErrorDay Element 334
Table 5-349 Item Description Specifications of ErrorHour Element 335
Table 5-350 Item Description Specifications of ErrorMinite Element 336
Table 5-351 Item Description Specifications of ErrorSecond Element 337
Table 5-352 Item Description Specifications of ErrorDayOfWeek Element 338
Table 5-353 Item Description Specifications of ErrorDetail** Element 339
Table 5-354 List of elements written in METHOD part 340
Table 5-355 List of elements written in COMM_IF_COMMAND part 340
Table 5-356 List of elements written in BLOCK_COMMAND part 340
Table 5-357 METHOD Specifications for conducting error check of equipment (if error code can be acquired) 341
Table 5-358 METHOD Specifications for conducting error check of equipment (when error code cannot be acquired) 342
Table 5-359 Example item description of elements of COMM_IF_COMMAND part 343
Table 5-360 Item Description Specifications of BLOCK_COMMAND Element 343
Table 5-361 Item description example of ErrorCode element 344
Table 5-362 Example of item description in the case of ErrorCode elements 345
Table 5-363 List of elements written in BLOCK_PARAMETER parts 345
Table 5-364 Item Description Specifications of P_TransIntervalMonitorTime Element of COMM_IF_PARAMETER Part 346
Table 5-365 Item Description Specifications of P_TransIntervalMonitorTime Element of BLOCK_PARAMETER Part 347
Table 5-366 Reserved words that indicated read/write of CRC32 values 348
Table 5-367 List of elements written in MESSAGE parts 349
Table 5-368 List of elements written in COM_IF_PARAMETER parts 349
Table 5-369 List of elements written in BLOCK_PARAMETER parts 349
Table 5-370 List of elements written in BLOCK_PARAMETER parts 349
Table 5-371 List of elements written in BLOCK_PARAMETER parts 350
Table 5-372 Item Description Specifications of SLMP_P_Safety_ParameterWrite Element 350
Table 5-373 Item Description Specifications of SLMP_P_Safety_EnableParameter Element 351
Table 5-374 Conventions of COMM_IF_PARAMETER parts 353
Table 5-375 List of elements written in P_ParamIDInfo* 353
Table 5-376 List of elements written in BLOCK_PARAMETER parts 353
Table 5-377 Item description specifications of COMM_IF_PARAMETER elements 354
Table 5-378 Item description specifications of elements of BLOCK_PARAMETER parts 355
Table 5-379 List of Elements Described in METHOD Part 357
Table 5-380 List of Elements Described in MESSAGE Part 357
Table 5-381 List of Elements Described in COMM_IF_COMMAND Part 357
Table 5-382 List of Elements Described in BLOCK_COMMAND Part 357
Table 5-383 List of Elements Described in COMMAND_ARGUMENT Part 358
Table 5-384 Item Description Specifications of P_GetEnergyInfo** Element 359
Table 5-385 Item Description Specifications of P_GetEnergyInfo** Element 361

Table 5-386 Item Description of GetEnergyInfoCommand Element 363
Table 5-387 Item Description Specifications of GetEnergyInfoCommand** Element 364
Table 5-388 Item Description Specifications of P_AverageCurrent Element 365
Table 5-389 Item Description Specifications of P_AverageVoltage Element 366
Table 5-390 Item Description Specifications of P_ElectricPower Element 367
Table 5-391 Item Description Specifications of P_ElectricEnergy Element 368
Table 5-392 Item Description Specifications of P_PowerDistributionTime Element 369
Table 5-393 Item Description Specifications of P_RunningTime Element 370
Table 5-394 Item Description Specifications of P_1_PhaseCurrent Element 371
Table 5-395 Item Description Specifications of P_2_PhaseCurrent Element 372
Table 5-396 Item Description Specifications of P_3_PhaseCurrent Element 373
Table 5-397 Item Description Specifications of P_1_2lineVoltage Element 374
Table 5-398 Item Description Specifications of P_2_3lineVoltage Element 375
Table 5-399 Item Description Specifications of P_3_1lineVoltage Element 376
Table 5-400 Item Description Specifications of P_1_PhaseCurrentDemand Element 377
Table 5-401 Item Description Specifications of P_2_PhaseCurrentDemand Element 378
Table 5-402 Item Description Specifications of P_3_PhaseCurrentDemand Element 379
Table 5-403 Item Description Specifications of P_ElectricPowerDemand Element 380
Table 5-404 Item Description Specifications of P_PowerFactor Element 381
Table 5-405 Item Description Specifications of P_Frequency Element 382
Table 5-406 Item Description Specifications of P_ReactiveEnergy Element 383
Table 5-407 Item Description Specifications of P_1_0PhaseVoltage Element 384
Table 5-408 Item Description Specifications of P_2_0PhaseVoltage Element 385
Table 5-409 Item Description Specifications of P_3_0PhaseVoltage Element 386
Table 5-410 List of Elements Described in COMM_IF_COMMAND Part 387
Table 5-411 List of Elements Described in BLOCK_COMMAND Part 387
Table 5-412 List of Elements Described in COMMAND_ARGUMENT Part 388
Table 5-413 List of Elements Described in ENUM Part 388
Table 5-414 Item Description of P_GetStatusCommand Element 389
Table 5-415 Item Description Specifications of Element 389
Table 5-416 Item Description Specifications of Element 390
Table 5-417 Item Description Specifications of ENUM Part Element 391
Table 5-418 List of Elements Written in MESSAGE Part 393
Table 5-419 List of Elements Written in COMM_IF_COMMAND Part 393
Table 5-420 List of Elements Written in BLOCK_COMMAND Part 393
Table 5-421 List of Elements Written in COMMAND_ARGUMENT Part 393
Table 5-422 List of Elements Written in ENUM Part 393
Table 5-423 List of Elements Written in COMMAND_ARGUMENT Part for Detailed Information Structure Definition 394
Table 5-424 List of Elements Written in STRUCT Part 394
Table 5-425 Item Description Specifications of P_GetDiagCodeInfo Element 394
Table 5-426 Item Description Specifications of GetDiagCodeInfoCommand Element 395
Table 5-427 Item Description Specifications of GetDiagCodeCommand Element 395
Table 5-428 Item Description Specifications of ErrorCode Element 396
Table 5-429 Item Description Specifications of Elements of ENUM Part 397
Table 5-430 Item Description Specifications of COMMAND_ARGUMENT Part Element for Detailed Information Structure Definition 398
Table 5-431 Item Description Specifications of STRUCT Part Element 399
Table 5-432 List of Elements Written in MESSAGE Part 401
Table 5-433 List of Elements Written in COMM_IF_PARAMETER Part 402
Table 5-434 Item Description Specifications of Element Written Using Only One Element 402

Table 5-435 Item Description Specifications of Element to Write "SEQ_TARGET" to TARGET 403
Table 5-436 Item Description Specifications of Elements Referenced from "SEQ_TARGET" Element 404
Table 5-437 Item Description Specifications of COMM_IF_PARAMETER Part Element 405
Table 5-438 List of Elements Written in COMM_IF_PARAMETER Part 406
Table 5-439 List of Elements Written in BLOCK_PARAMETER Part 406
Table 5-440 Item Description Specifications of P_TransIntervalMonitorTime Element in COMM_IF_PARAMETER Part 407
Table 5-441 Item Description Specifications of P_SafetyCertCode Element in COMM_IF_PARAMETER Part 408
Table 5-442 Item Description Specifications of P_TransIntervalMonitorTime Element in BLOCK_PARAMETER Part 409
Table 5-443 Item Description Specifications of P_SafetyCertCode Element in BLOCK_PARAMETER Part 410
Table 5-444 Reserved Words to Indicate Reading and Writing CRC32 Value 411
Table 5-445 List of Elements Described in MESSAGE Part 412
Table 5-446 List of Elements Described in COMM_IF_PARAMETER Part.... 412
Table 5-447 List of Elements Described in BLOCK_PARAMETER Part.... 412
Table 5-448 List of Elements Described in COMM_IF_COMMAND Part.... 413
Table 5-449 List of Elements Described in BLOCK_COMMAND Part 413
Table 5-450 Item Description Specifications of SLMP_P_Safety_ParameterWrite Element 413
Table 5-451 Item Description Specifications of COMM_IF_PARAMETER Part Element 414
Table 5-452 Item Description Specifications of BLOCK_PARAMETER Part Element 415
Table 5-453 Item Description Specifications of SLMP_P_Safety_EnableParameter Element 416
Table 5-454 Item Description Specifications of COMM_IF_COMMAND Part Element.... 417
Table 5-455 Item Description Specifications of BLOCK_COMMAND Part Element.... 417
Table 6-1 Information for Identifying CSP+ File and Management Method by Utility Software 420
Table 8-1 CSV File Format 432
Table 8-2 Item List 433

1. FOREWORD

This document is the Control & Communication System Profile Specifications.

It contains conventions for CSP+ files usable in application software that launches, operates, and maintains CC-Link family connection units (hereinafter referred to as "utility software")

The CC-Link family connection unit written in this document is a unit connected to the network by the following connection interface.

(1) CC-Link

(2) CC-Link IE Field Network

(3) Ethernet

(4) CC-Link IE TSN

Here, the Ethernet connection unit is a unit possessing an Ethernet physical layer interface, outfitted with TCP/IP protocol stock, that performs CC-Link IE Field Network Basic communication or conducts SLMP communication using TCP or UDP packets.

The version of Control & Communication Profile Specifications written in this document (hereinafter referred to as CSP+ Specifications Version) is version 3.1. Usable conventions varies depending on CSP+ Specifications Version. See Chapter 9 for details.

2. OVERVIEW

2.1 What is CSP+?

Control & Communication Profile (CSP+) are specifications for for writing the necessary information to start up, operate, and maintain CC-Link Family Connection units.

In the past, the CC-Link system profile (CSP) has been used to describe such information in relation to CC-Link connected modules. Since the enactment of CSP specifications, however, the need for profile specifications that offer a higher level of descriptiveness has increased due to reasons such as the following:

(1) Module diversification
While in the past single-function products with minimal setup were the mainstream, there has been an increase in highly functional modules capable of complex setup and control in recent years.

(2) Utility software application diversification
Not only has module input/output and parameter setup become increasingly diversified, but so have the objectives of utility software usage, such as diagnostics and energy management (future concept).

(3) Network diversification
Protocols for connecting modules, including CC-Link, CC-Link IE Field Network, SLMP (Seamless Message Protocol), and CC-Link IE TSN have become increasingly diversified.

In this context, CSP+ is now to be defined as the protocol specifications for CC-Link family connected modules.

2.2 FEATURES

This section describes the features of CSP+.

(1) Integrates the utility software environment
With CSP+, CSP+ files are imported into utility software when starting, running, and maintaining a module, making it possible to manage all modules connected to the CC-Link family from a single utility software. If the development vendor of the module creates a CSP+ file for the module to be developed, there is no need to create individual utility software.

(2) Integrates different protocol profile specifications
CSP+ allows profile descriptions to be executed using the same format for modules that are connected via different communication protocols, such as CC-Link connected modules and CC-Link IE Field Network connected modules.

(3) Offers high descriptiveness, achieving utility software user-friendliness
CSP+ improves the user-friendliness of utility software with its high descriptiveness, such as its labels applicable with ladders and groupings for collectively displaying parameters by function on utility software screens.

(4) Offers application compatible descriptions
CSP+ offers application compatible profile descriptions, such as profile descriptions for diagnostics and energy management (future concept), allowing the display of special screens with layouts specialized for each application via utility software.

2.3 Flow of CSP+ File Utilization

This section describes the flow of CSP+ file utilization.

(1) Creating and distributing a CSP+ file
When a module development vendor develops a CC-Link family connected module, the vendor creates a CSP+ file for the module to be developed, and distributes that file to the user. When creating the CSP+ file, refer to this document and create the file so that it is compliant with CSP+ specifications.

(2) Importing the CSP+ file
Next, the CSP+ file is imported into the utility software used for starting, running, and maintaining the module to be utilized. To utilize CSP+ with utility software, the utility software must have a function for importing CSP+. CSP+ compatible software includes, for example, a parameter setup tool, monitor tool, diagnostics tool, and energy management tool. The vendor that develops the master module or the vendor that develops applications for diagnostics, energy management, or the like, provides the CSP+ compatible utility software.

(3) Utilizing the utility software
The utility software is then utilized to start, run, and maintain the target module. The functions provided differ for each utility software. Select the utility software in accordance with the purpose of use.

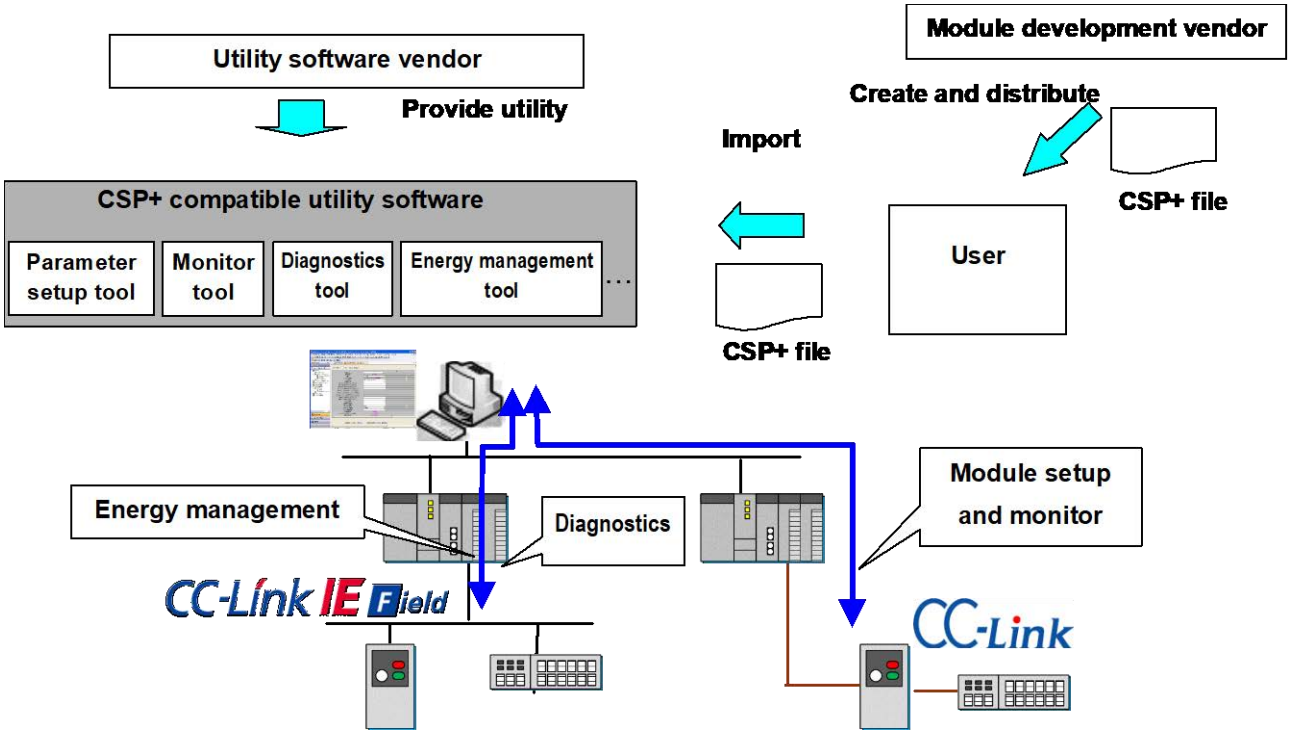


Figure 2-1 CC-Link Family System Profile (CSP+)

3. GENERAL CONCEPT OF CSP+ FILE DESCRIPTION SPECIFICATIONS

This chapter describes the basic concept of CSP+ file creation.

3.1 Model of CC-Link Family Connected Module

With CSP+, CC-Link family connected modules are separated into virtual field devices that represent communication function sections and virtual control devices that represent module-specific sections. These devices are then modeled and respectively described to support a variety of communication. At this time, the input, output, parameter, and command access of the virtual control devices from the utility software via the network is considered to be respectively executed via the outputs, inputs, parameters, and commands (see Section 3.2) of the corresponding network virtual field devices.

The descriptions related to the virtual control devices can be standardized between modules having the same module-specific functions but different communication protocols by separately describing communication protocol dependent virtual field devices and module-specific functions.

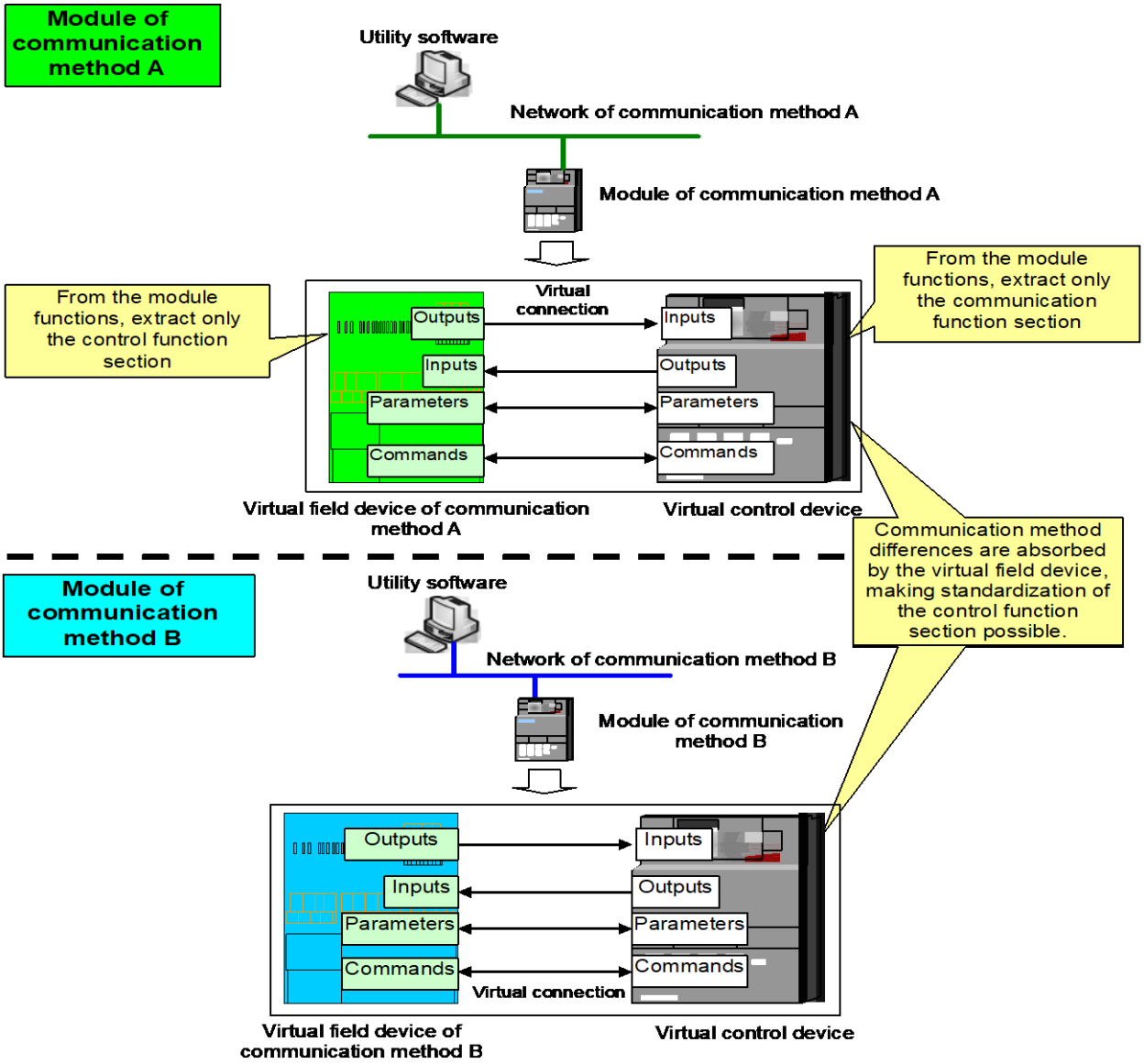


Figure 3-1 Model of CSP+ Module

3.2 Models by Module Access Method

This chapter describes a CSP+ module model for each module access method.

Below are described the items common to all module access modeling.

(1) Input/Output
When module input/output contacts or input/output areas are directly allocated for reading and writing with cyclic transmission remote inputs/outputs or remote registers, or when transient transmission is used to perform reading and writing with module input/output contacts or input/output areas, input or output modeling is performed.

(2) Parameters
When part of the cyclic transmission area is temporarily used to send and receive module parameters, or when transient transmission is used to send and receive module parameters, parameter modeling is performed.

(3) Commands
When part of the cyclic transmission area is temporarily used to request module commands, or when transient transmission is used to request module commands, command modeling is performed.

3.2.1 Access by cyclic transmission

This section describes how the previously described common items are expressed in virtual control devices and virtual field devices for module access via cyclic transmission.

(1) Input/Output
In a case where access via cyclic transmission is described for input/output, sometimes module bit-unit data (input contacts, output contacts, etc.) is allocated in a fixed manner to cyclic transmission remote input/output (RX, RY) and used, and sometimes byte-unit or word-unit data is allocated in a fixed manner to remote registers (RWr, RWw) and used.
At this time, the information input to the master station is described as virtual field device input, with output further described for virtual control devices and information related to virtual control device output further described for virtual field device input. Similarly, information output from the master station is described as virtual field device output, with input further described for virtual control devices and information related to virtual control device input further described for virtual field device output.
Figure 3-2 shows an example of a case where access via cyclic transmission is described for input and output. Figure 3-2 is an example of a case where RY0 is associated with output contact 1, and RX0 is associated with input contact 1. As shown in the figure, RY0 and RX0 of the module are respectively described as the output and input of the virtual field device, and output contact 1 and input contact 1 of the module are respectively described as the input and output of the virtual control device.

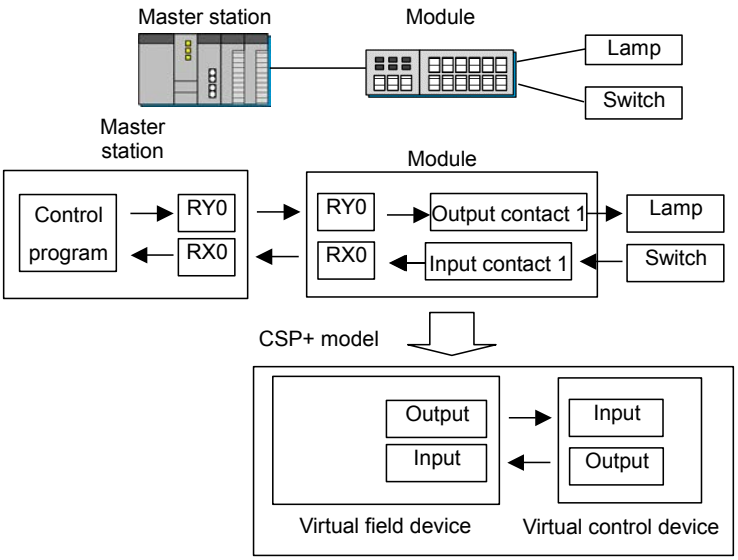


Figure 3-2 Example of Input/Output by Cyclic Transmission

(2) Parameters

In a case where access via cyclic transmission is described for parameters, sometimes a part of the cyclic transmission remote register and remote input/output area is temporarily used to transmit and receive module parameters.

At this time, parameter related information is described as virtual field device parameters, with parameters further described for virtual control devices and information related to virtual control device parameters further described for virtual field device parameters. In addition, information related to parameter transmission/reception procedures is described as virtual field device methods, with information related to virtual field device parameters described in the method.

Note that "a part of the cyclic transmission remote register and remote input/output area is temporarily used to transmit and receive module parameters" refers to a case where parameter reading and writing is expressed by a procedure such as the following, for example (the following is an example of parameter writing):

[1] The master station writes a parameter value to RWw1.

[2] The master station turns RY1 ON.

[3] The module detects RY1 ON.

[4] The module reads the parameter value from RWw1.

[5] The module reflects the parameter value into the internal memory of the module.

Figure 3-3 shows an example of a case where access via cyclic transmission is described for parameters. Figure 3-3 is an example of a case where a module has the parameters parameter 1 and parameter 2, and RWw1 and RY1 are used in the parameter transmission/reception procedure. As shown in Figure 3-3, parameter 1 and parameter 2 of the module are described as the virtual field device parameter and virtual control device parameter, and the parameter transmission/reception procedure is described as the method.

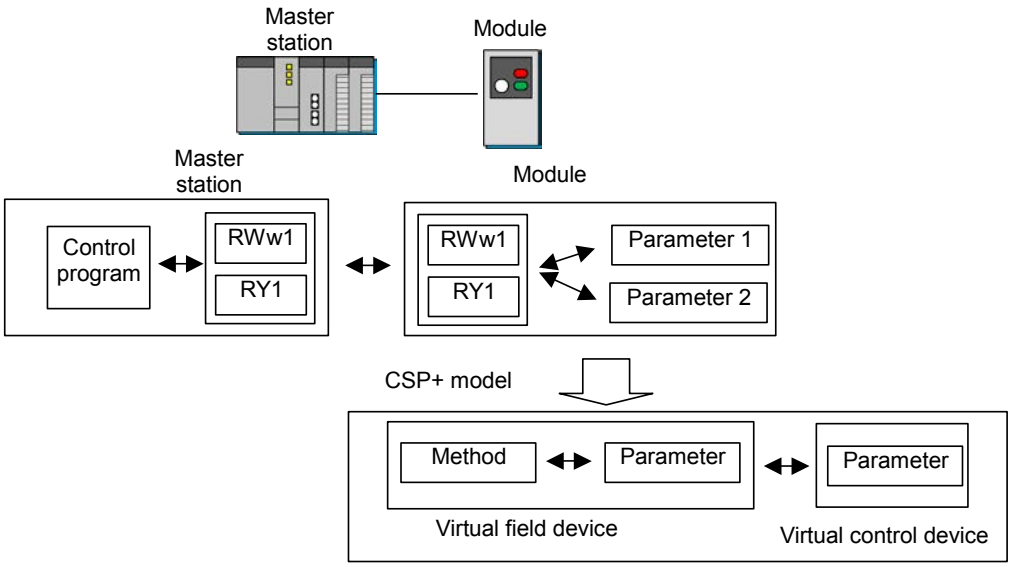


Figure 3-3 Example of Parameter Transmission/Reception by Cyclic Transmission

(3) Commands

In a case where access via cyclic transmission is described for commands, sometimes a part of the cyclic transmission remote register and remote input/output area is temporarily used for command requests to modules.

At this time, command related information is described as virtual field device commands, with commands further described for virtual control devices and information related to virtual control device commands further described for virtual field device commands. In addition, information related to command request procedures is described as virtual field device methods, with information related to virtual field device commands described in the method.

Note that "a part of the cyclic transmission remote register and remote input/output area temporarily used for command requests to modules" refers to a case where command execution requests held by devices are expressed by a procedure such as the following, for example:

[1] The master station writes a command code to RWw2.

[2] The master station turns RY2 ON.

[3] The module detects RY2 ON.

[4] The module reads the command code from RWw2.

[5] The module executes the command.

Figure 3-4 shows an example of a case where access by cyclic transmission is described for commands. Figure 3-4 is an example of a case where the module can execute the command entitled command 1, and RWw2 and RY2 are used in the command request procedure. As shown in Figure 3-4, command 1 of the module is described as a virtual field device command and virtual control device command, and the command request procedure is described as the method.

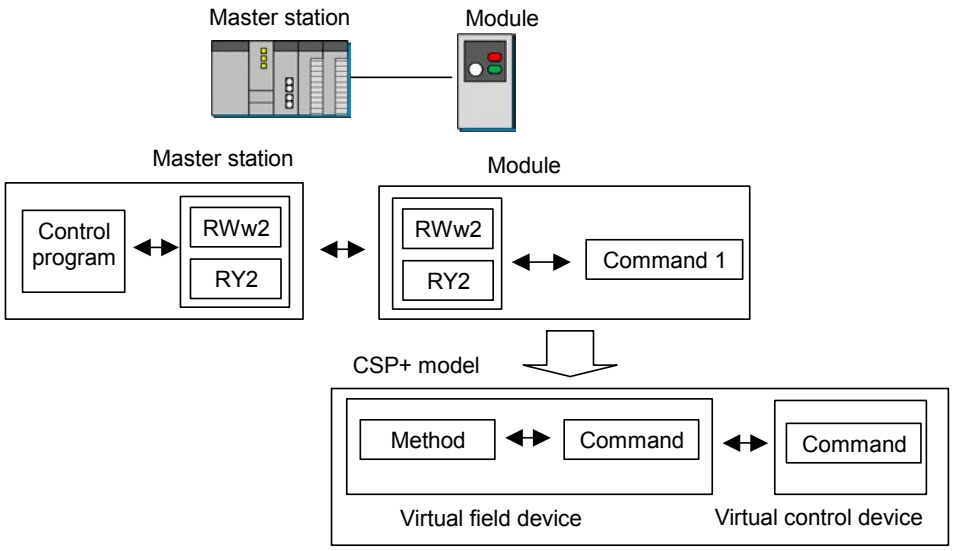


Figure 3-4 Example of Command Transmission by Cyclic Transmission

3.2.2 Access by transient transmission

This section describes how the previously described common items are expressed in virtual control devices and virtual field devices for module access via transient transmission.

Note that, with CSP+ specifications, access via SLMP can be described for transient transmission.

(1) Input/Output

In a case where access via transient transmission is described for input or output, sometimes SLMP is used for input/output control.

At this time, the information input to the master station is described as virtual field device input, with output further described for virtual control devices and information related to virtual control device output further described for virtual field device input. In addition, SLMP command information used by the master station for data reading is described as virtual field device messages. Similarly, information outputted from the master station is described as virtual field device output, with input further described for virtual control devices and information related to virtual control device input further described for virtual field device output. Further, the SLMP command information used for data writing is similarly described as virtual field device messages.

Figure 3-5 shows an example of a case where access via transient transmission is described for input and output. Figure 3-5 is an example of a case where output contact 1 and input contact 1 of the module are transmitted and received via SLMP messages. As shown in Figure 3-5, output contact 1 and input contact 1 of the module are respectively described as the output and input of the virtual field device and the input and output of the virtual control device, and the SLMP command information to be used is described as a message.

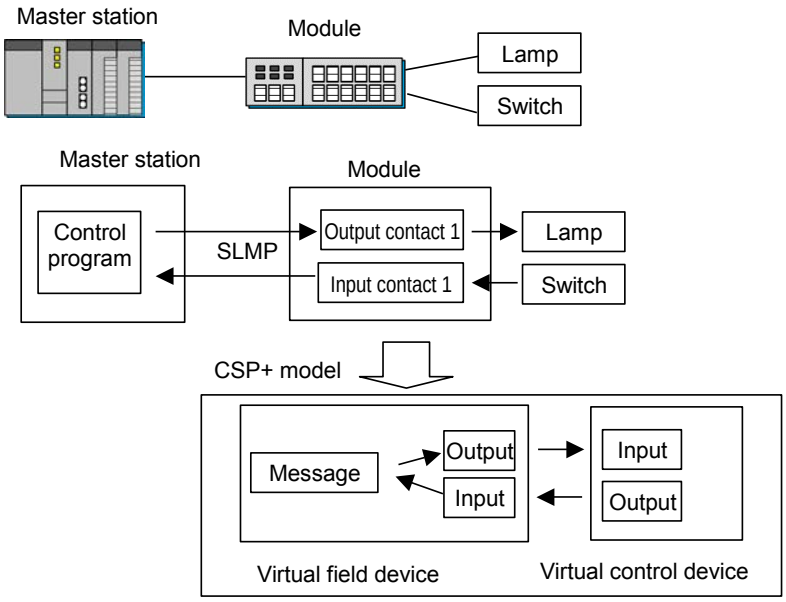


Figure 3-5 Input/Output by SLMP

(2) Parameters

In a case where access via transient transmission is described for parameters, sometimes SLMP is used for module parameter transmission and reception.

At this time, parameter related information is described as virtual field device parameters, with parameters further described for virtual control devices and information related to the virtual control device parameters further described for virtual field device parameters. In addition, SLMP command information used for parameter reading and writing is described as virtual field device messages, with information related to virtual field device parameters described in the messages.

Figure 3-6 shows an example of a case where access via transient transmission is described for parameters. Figure 3-6 is an example of a case where a module has the parameters parameter 1 and parameter 2, and transmission/reception is performed via SLMP messages. As shown in Figure 3-6, parameter 1 and parameter 2 of the module are described as the virtual field device parameter and virtual control device parameter, and the SLMP command information to be used is described as the message.

When parts or elements with defined applications shown in Section 5.5 are used, the message part may not need to be described (see Section 5.5.7). In this case, parameters are transmitted or received using the method which is determined beforehand corresponding to the specific part name or element name (such as specific SLMP command).

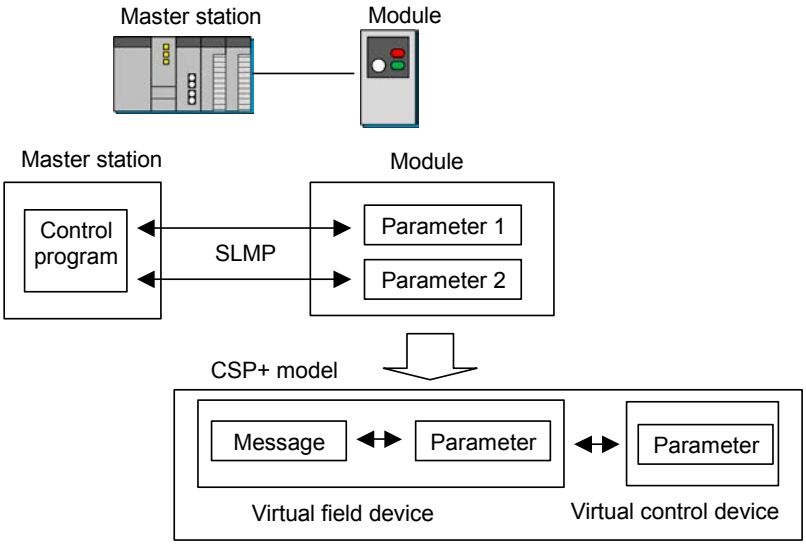


Figure 3-6 Parameter Transmission/Reception by SLMP

(3) Commands

In a case where access via transient transmission is described for commands, sometimes SLMP is used for command requests to modules.

At this time, command related information is described as virtual field device commands, with commands further described for virtual control devices and information related to virtual control device commands further described for virtual field device commands. In addition, SLMP command information used for command requests is described as virtual field device messages, with information related to virtual field device commands described in the message.

Figure 3-7 shows an example of a case where access by transient transmission is described for commands. Figure 3-7 is an example of a case where the module can execute the command entitled command 1, and the request is made via an SLMP message. As shown in Figure 3-7, command 1 of the module is described as a virtual field device command and virtual control device command, and the SLMP command information to be used is described as a message.

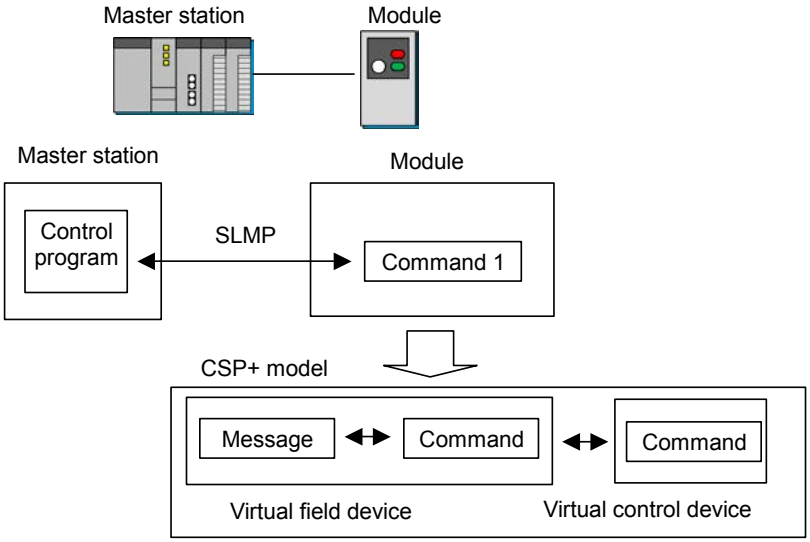


Figure 3-7 Command Transmission by SLMP

3.3 Models by Module Form

This section describes a module model for CSP+ for each module form. Each model describes only the information for one CSP+ file. Note that, for explanatory purposes, method and message descriptions are omitted.

(1) Module connected to a single network

Figure 3-8 shows a model of a module connected to a single network. With CSP+, output, input, parameter, and command descriptions are provided for the virtual field device, and input, output, parameter, and command descriptions are provided for the virtual control device.

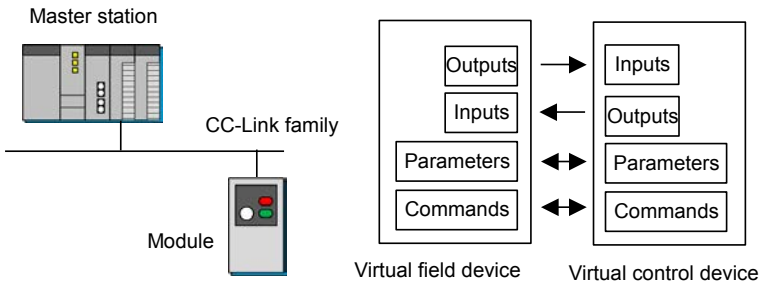


Figure 3-8 Model of Module Connected to Single Network

(2) Module used while switching between multiple networks

With a module that is used while switching between multiple networks, such as CC-Link and the CC-Link IE Field Network, separate CSP+ files are created for each network. Distinguish the files using the "arbitrary string" area in the CSP+ file name so that the file names are not duplicated (For descriptions of arbitrary string in the CSP+ file name, see Section 4.1.1).

Figure 3-9 shows a model used while switching between multiple networks.

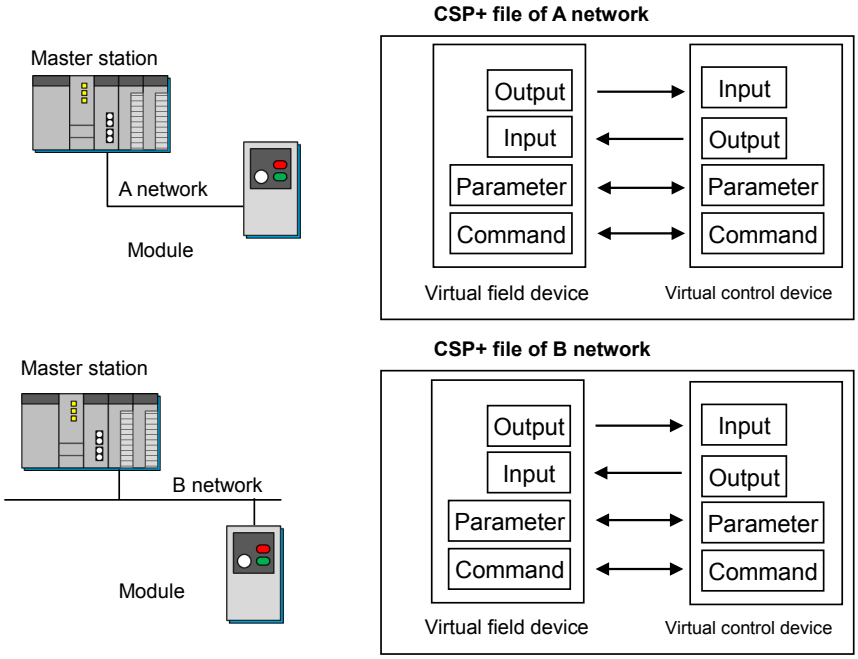


Figure 3-9 Model of Module Used While Switching Between Multiple Networks

(3) Module used with multiple modules connected

When a module is used with several modules connected to a communication head module, a CSP+ file must be created that describes only the communication head module using a basic form of the CSP+ file corresponding to the module. Aside from this, however, it is possible to write and utilize a CSP+ file for a configuration that includes the other modules connected to the communication head module.

Figure 3-10 shows a model of a case where only the communication head module is described in the CSP+ file, and Figure 3-11 and Figure 3-12 show models of a case where a CSP+ file is created for a configuration that includes the other modules connected to the communication head module. These examples are for a scenario in which there are two other modules (module A and module B) connected to the communication head module.

When a CSP+ file is written for the communication head module only, the model is the same as that for a single module.

As shown in Figure 3-10, inputs, outputs, parameters, and commands used in the communication head module are described as descriptions of the virtual control device.

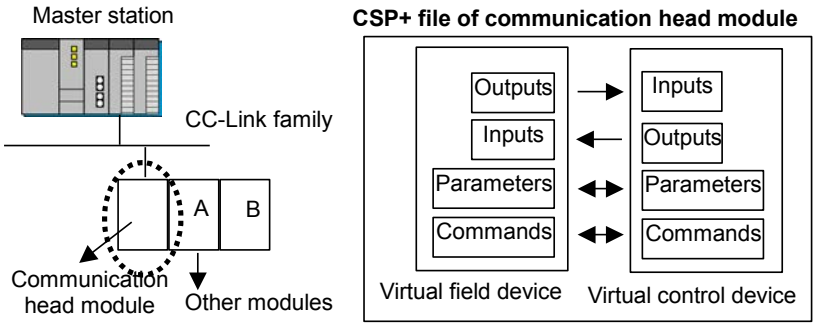


Figure 3-10 Model of Module When Only Communication Head Module Is Described

In a case where a CSP+ file is written for a configuration that includes the other modules connected to the communication head module, the file can be written with each of the modules regarded as separate virtual control devices. However, writing a separate virtual control device for each module in this manner is not required; the virtual control devices for the communication head module and module A may be described together as one.

Figure 3-11 shows an example of a case where the other modules connected with the communication head module are described as separate virtual devices. The example in Figure 3-11 is of a case where there are two other modules connected: module A and module B. As a result, three virtual control devices are described: communication head module, module A, and module B. In this case, each of the elements described in the outputs, inputs, parameters, and commands of the virtual field device are associated with the input, output, parameter, and command elements of the virtual control device of the communication head module and the virtual control device of module A.

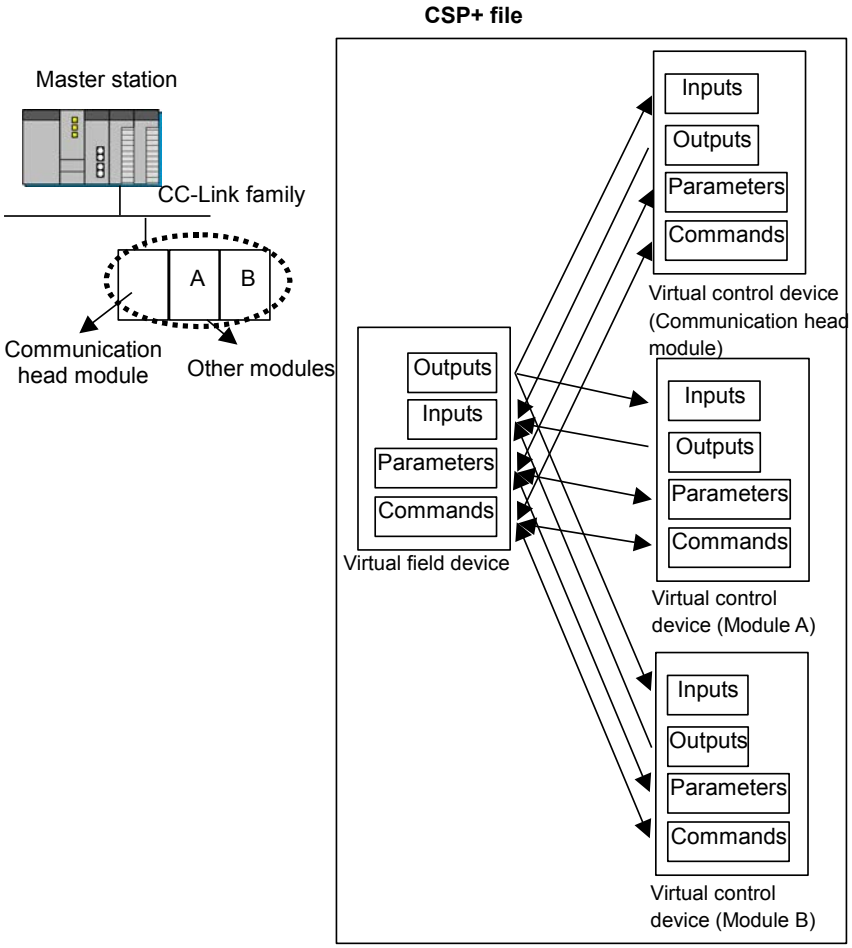


Figure 3-11 Model of Module When CSP+ File Is Written Based on a Configuration that Includes Connected Modules (When Virtual Control Devices Are Separately Described for Communication Head Module and Other Modules)

Figure 3-12 shows an example of a case where the communication head module and the other connected modules are described together as one. The example in Figure 3-12 is of a scenario where there are two other connected modules: module A and module B.

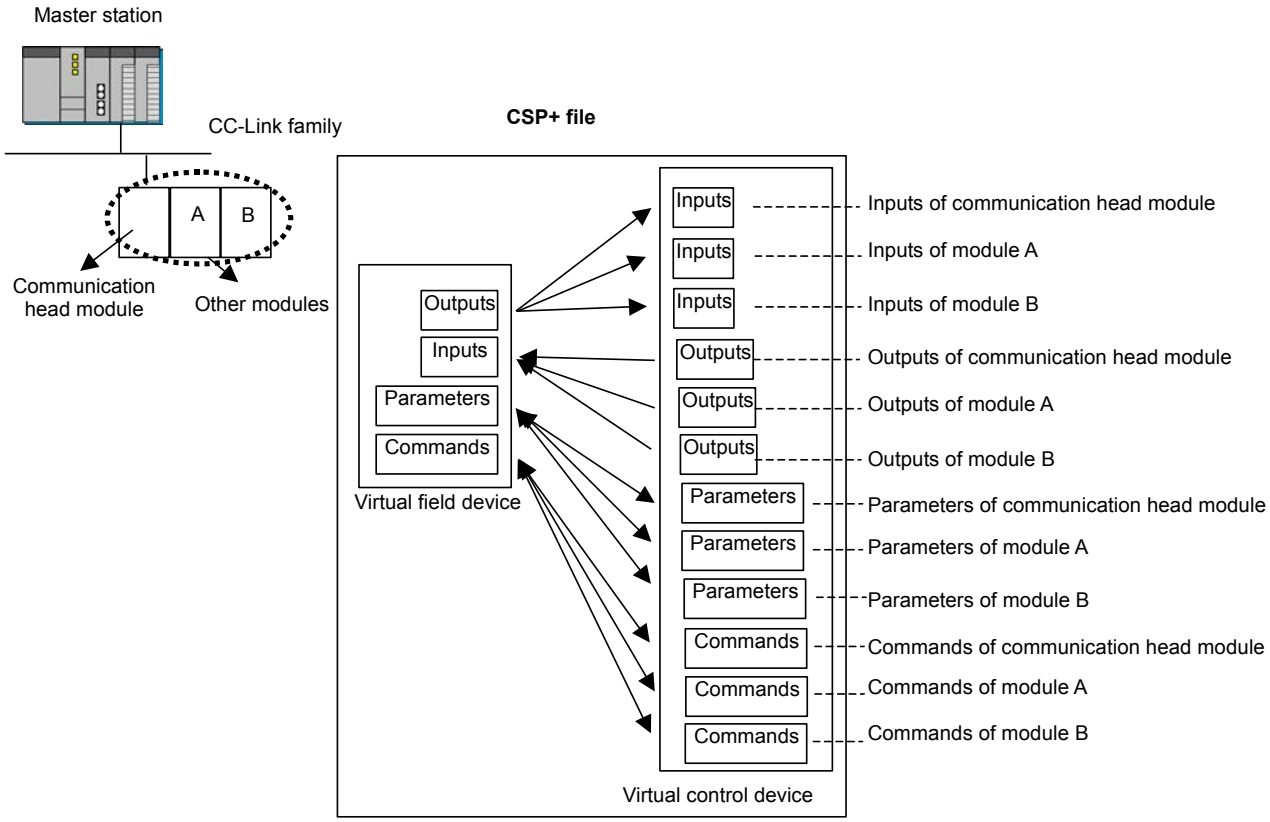


Figure 3-12 Model of a Module When a CSP+ File Is Written Based on a Configuration that Includes Connected Modules (When the Virtual Control Devices of the Communication Head Module and Other Modules Are Described Together)

4. CSP+ FILE STRUCTURE AND DESCRIPTION SPECIFICATIONS

4.1 CSP+ File Management Specifications

CSP+ files can be managed together in multiple along with CSP+ file related files (icon files, graphics files) as a compressed file.

If the utility software supports compressed files, the CSP+ files can then be imported by specifying the compressed file. With this in mind, note that, in this document, importing a CSP+ file can also be read as importing a compressed file.

4.1.1 CSP+ file name

A CSP+ file is named as follows:

VendorCode_ModelName(DeviceConfigurationID)_DeviceVersion_LanguageInformation[_ArbitraryString].cssp

- Description in parentheses "( )" and "[ ]" may be omitted.
- When using device configuration IDs, write the round parentheses "( )".
- When using arbitrary strings, there is no need to write the square brackets "[ ]".

The vendor code is the code of the vendor that manufactured the module. The code is always expressed using four digits. For example, "0x0000" is written as "0x0000" and not abbreviated to "0x0". In addition, "0x" which indicates a hexadecimal number must also be written. Use the value written in the element "Vendor Code" of the part DEVICE_INFO described later.

The model name is the model name of the module. Use the string written in the element "Device Model" of the part Device_INFO described later.

The device configuration ID is used for identification purposes when multiple profiles are defined by the device configuration. When the device configuration ID is defined, add the ID after the model name. Use the string written in the element "Device Configuration ID" of the part Device_INFO described later. If a device configuration ID is not defined, the string in the "(Device Configuration ID)" section is not required.

The device version is the device version of the module. Use the string written in the element "Display Version Value" of the part Device_INFO described later. However, if the element "Display Version Value" is omitted, use the string written in the element "Version" of the part DEVICE_INFO.

The language information is the information that indicates the language supported by the CSP+ file description. Use the string written in the element "Language" of the part FILE_INFO described later.

An arbitrary string is used to prevent file names from being duplicated when CSP+ files are created for each network with the devices that can be connected to different networks.

The following describes the reference examples for the arbitrary string.

- CC-Link : CCLink
- CC-Link IE Field : CCLinkIEField
- Ethernet *Including Basic : Ethernet
- CC-Link IE TSN : CCLinkIETSN

All of this information is used for the file name as well. If an unpermitted character (such as "/") is used in the file name, that character is omitted.

The vendor code, model name, device version, and language information are all described in the CSP+ file as module information and file information as well. For details, see Chapter 5.

4.1.2 Data configuration with compression management

This section describes the specifications of compressed files.

Usage of compressed files allows you to manage multiple CSP+ files and CSP+ file associated icon files, graphics files, and object dictionary files together. Compress and manage compressed files in zip format. Creation of a compressed file requires compliance with the following specifications.

- The files and folders within a compressed file need to be configured as shown in Figure 4-1. Compress the files by specifying one or more CSP+ files in the upper most level of the compressed file, and a folder that stores graphics ("Folder" in Figure 4-1), without preparing an upper-most folder that includes all of the files to be compressed.
- The names of the CSP+ files must comply with the specifications indicated in Section 4.1.1.
- The graphics folder name must be the vendor code. Note the following:
  - Always indicate the vendor code using a 4-digit number. For example, write "0x0000" as "0x0000" and not the abbreviated "0x0".
  - "0x", which indicates a hexadecimal number, needs to be written as well.
  - Use the value written in the element "VendorCode" of the part DEVICE_INFO described later.
  - If graphics files are not required, a graphics folder is not necessary.
- Each file name needs to be the file name written as the icon file name, graphics file name, and object dictionary file name in the CSP+ file. Write the icon file name in the element "IconFileName" of the part DEVICE_INFO described later. Write the graphics file name in the element "GraphicsFileName" in the part DEVICE_INFO described later. Write the object dictionary file name in the element "ObjectDictionaryFileName" of the COMM_IF_INFO part described later. (For icon file name, graphics file name, and object dictionary file name details, see Chapter 5.)
- All of the icon file, graphics file, and object dictionary file must be stored in the same folder ("Folder" in Figure 4-1).
- CSP+ files from multiple vendors cannot be managed together.
- The names of all files and folders to be compressed must be written using single-byte alphanumeric characters and symbols.
- The name of the compressed file can be freely determined by the vendor.

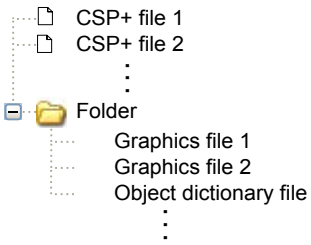


Figure 4-1 Configuration within Compressed File

[Note] The object dictionary file is required when the CAN compatible device of CC-Link IE TSN is used. For details, see Section 4.2.8.4.

4.2 CSP+ File Structure

CSP+ files are written using XML.

4.2.1 Definitions of sections within a CSP+ file

A CSP+ file comprises a FILE section, DEVICE section, BLOCK section, and COMM_IF section. Figure 4-2 shows the structure of a CSP+ file.

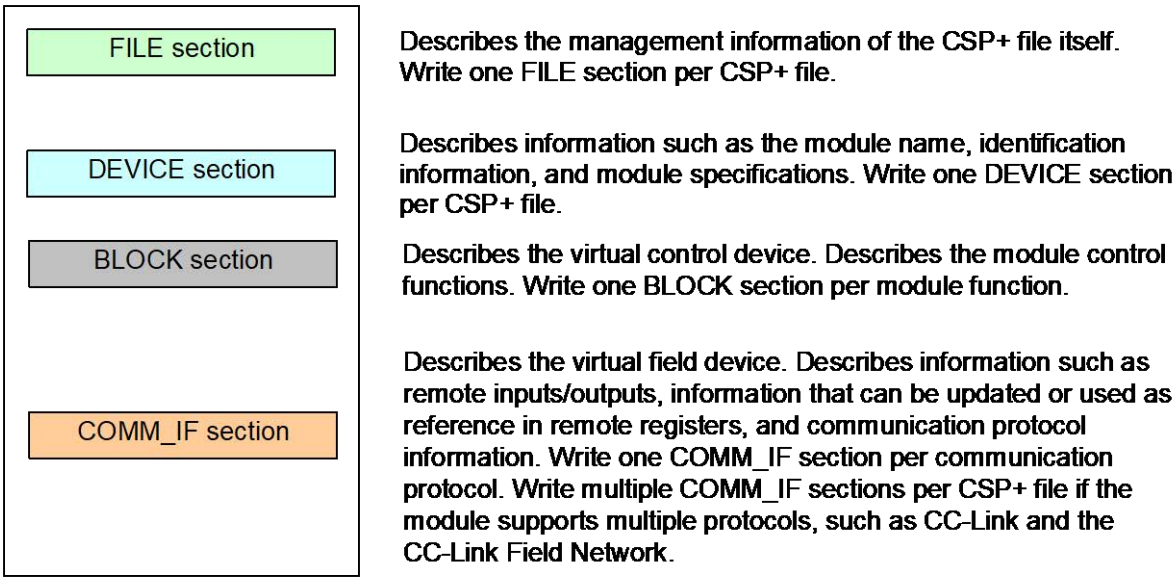


Figure 4-2 CSP+ File Structure

4.2.2 Definitions of parts within a CSP+ file section

Each section of a CSP+ file comprises multiple parts. Figure 4-3 shows the structure within a section.

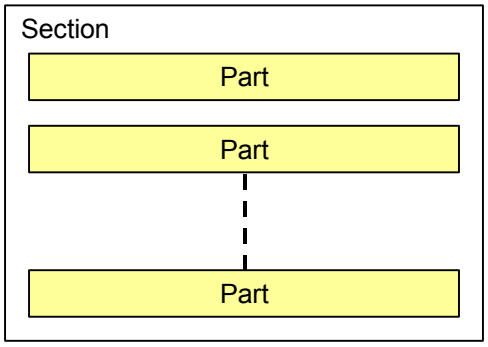


Figure 4-3 Section Structure

The configuration of a part is determined for each section. Table 4-1 to Table 4-4 list the parts that make up each section.

Table 4-1 Parts of a FILE Section

No.	Part	Information Described
1.	FILE_INFO (See Section 5.1.1.)	• CSP+ file management information (Examples of elements: File creation date, language information, file version)

One FILE_INFO part needs to be written in a FILE section.

Table 4-2 Parts of a DEVICE Section

No.	Part	Information Described
1.	DEVICE_INFO (See Section 5.2.1.)	• Module identification information (Examples of elements: Vendor name, model name) • Module product information (Examples of elements: Specification information, power supply voltage, current consumption, external dimensions)

One DEVICE_INFO part needs to be written in a DEVICE section.

Table 4-3 Parts of a COMM_IF Section

No.	Part	Information Described
1.	COMM_IF_INFO (See Section 5.3.1.)	• Communication function identification information (Examples of elements: Vendor name, communication method) • Communication function specifications (Examples of elements: Station type, CC-Link version, number of occupied stations)
2.	COMM_IF_OUTPUT (See Section 5.3.2.)	• Reference (address assignment) information for "Control function input information" • Communication function output information (Examples of elements: CC-Link module error reset request flag)
3.	COMM_IF_INPUT (See Section 5.3.3.)	• Reference (address assignment) information for "Control function output information" • Communication function input information (Examples of elements: CC-Link module remote ready)
4.	COMM_IF_PARAMETER (See Section 5.3.4.)	• Reference information for "Control function parameter information" • Communication function parameter information (Examples of elements: Operation at communication reset)
5.	COMM_IF_COMMAND (See Section 5.3.5.)	• Reference information for "Control function command information" • Communication function dedicated command information (Examples of elements: Communication reset)
6.	METHOD (See Section 5.3.6.)	• Communication service information for "Procedures and data area specification" Used for communication services, handshakes, and reading/writing. Register information.
7.	MESSAGE (See Section 5.3.7.)	• Communication service information for "Data format specification" Communication service target, and request and response data format information
8.	STRUCT (See Section 5.3.8.)	Remote register (RWw/RWr) bit assignments
9.	ENUM (See Section 5.3.9.)	Setting range options
10.	COMMAND_ARGUMENT (See Section 5.3.10.)	Argument information for the part COMM_IF_COMMAND

The COMM_IF section includes parts that may be omitted.
One COMM_IF_INFO part needs to be written in the COMM_IF section.
All parts other than the COMM_IF_INFO part can be written in multiple within the section.

Table 4-4 Parts of a BLOCK Section

No.	Part	Information Described
1.	BLOCK_INFO (See Section 5.4.1.)	• Control function identification information (Examples of elements: Vendor name)
2.	BLOCK_INPUT (See Section 5.4.2.)	• Control function input information (Examples of elements: Output current value, forward run command)
3.	BLOCK_OUTPUT (See Section 5.4.3.)	• Control function output information (Examples of elements: Voltage input value, in forward run)
4.	BLOCK_PARAMETER (See Section 5.4.4.)	• Control function parameter information (Examples of elements: Range settings, upper limit frequency, lower limit frequency)
5.	BLOCK_COMMAND (See Section 5.4.5.)	• Control function command information (Examples of elements: Reset, parameter batch clear, data acquisition at error)
6.	STRUCT (See Section 5.4.6.)	Bit assignments of the part COMMAND_ARGUMENT
7.	ENUM (See Section 5.4.7.)	Setting range options
8.	COMMAND_ARGUMENT (See Section 5.4.8.)	Argument information for the part BLOCK_COMMAND

The BLOCK section includes parts that may be omitted.
One BLOCK_INFO part needs to be written in the COMM_IF section.
All parts other than the BLOCK_INFO part can be written in multiple within the section.

4.2.3 Definitions of elements and items of CSP+ parts

Each part of a section of a CSP+ file is made up of one or more elements. Each element is made up of one or more items. Figure 4-4 shows the relationship between elements, which make up a part, and items, which make up an element.

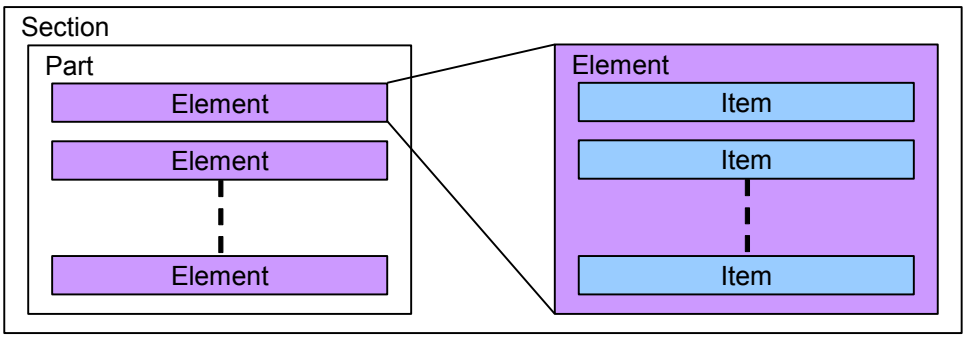


Figure 4-4 Part Structure

Table 4-5 and Table 4-6 list the items of each part.

Table 4-5 Items of Elements of Parts

Section Type	Part Type \ Item Name	LABEL	LABEL2	CATEGORY	NAME	DATATYPE	DATA	DEFAULT	RANGE	MIN_INC	ENG_UNIT	ACCESS	ASSIGN	UI_ATTRIBUTE	WRITE_ORDER	ARGUMENT	OFFSET	CODE	RELATED_ELE	CONDITIONx	RESULTx	REF	COMMENT
FILE	FILE_INFO	●	○	○	○	●	●																
DEVICE	DEVICE_INFO	●	○	○	○	●	●																
COMM_IF	COMM_IF_INFO	●	○	○	○	●	●																
	COMM_IF_INPUT	●	○	○	○	○		○	○	○	○	○	○	○								○	○
	COMM_IF_OUTPUT	●	○	○	○	○		○	○	○	○	○	○	○								○	○
	COMM_IF_PARAMETER	●	○	○	○	○		○	○	○	○	○	○	○	○							○	○
	COMM_IF_COMMAND	●	○	○	○											○						○	○
	METHOD	Described in separate table																					
	MESSAGE	Described in separate table																					
	STRUCT	●	○	○	○	○		○	○	○	○						●					○	○
	ENUM	●	○	○	●													●	○				○
	COMMAND_ARGUMENT	●	○	○	●	●		○	○	○	○	●	○										○
BLOCK	BLOCK_INFO	●	○	○	○	●	●																
	BLOCK_INPUT	●	○	○	●	●		○	○	○	○	○		○									○
	BLOCK_OUTPUT	●	○	○	●	●		○	○	○	○	○		○									○
	BLOCK_PARAMETER	●	○	○	●	●		○	○	○	○	●		○	○								○
	BLOCK_COMMAND	●	○	○	●											○							○
	STRUCT	●	○	○	●	●		○	○	○	○						●						○
	ENUM	●	○	○	●													●	○				○
	COMMAND_ARGUMENT	●	○	○	●	●		○	○	○	○	●	○										○
	CONDITION																			●	●		

○:Optional
●:Required

Table 4-6 Items of Elements of METHOD and MESSAGE Parts

Part Name \ Item Name	LABEL	LABEL2	CATEGORY	NAME	TARGET	METHOD_TYPE	WRITE_REGISTER	WRITE_DATA	WRITE_DATATYPE	READ_REGISTER	READ_DATA	READ_DATATYPE	INTERROCK	REQ_FLAG	END_CONDITION	ERR_CONDITION	ERR_REGISTER	ERR_CODE_RANGE	RELATED_METHOD	MESSAGE_TYPE	REQUEST_TYPE	REQUEST_DATA	REQUEST_DATATYPE	RESPONSE_TYPE	RESPONSE_DATA	RESPONSE_DATATYPE	ERR_YPE	RELATED_MESSAGE	COMMENT
METHOD	●	○	○	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○										○
MESSAGE	●	○	○	●	●													○		●	●	○	○	○	○	○	○	○	○

Table 4-7 shows a list of items. The items available for each part are described in Chapter 5.

Table 4-7 Item List

No.	Item Name	Overview	Description Location
1.	ACCESS	Describes access attributes.	Section 4.3.1.1
2.	ARGUMENT	Describes label names that indicate arguments.	Section 4.3.1.2
3.	ASSIGN	Describes assignments.	Section 4.3.1.3
4.	CATEGORY	Describes the group name with which the element is affiliated.	Section 4.3.1.4
5.	CODE	Describes codes.	Section 4.3.1.5
6.	COMMENT	Describes comments.	Section 4.3.1.6
7.	CONDITIONx	Writes the range of conditions for determination of conditional expressions.	Section 4.3.1.7
8.	CONDITIONXVARIABLE	Writes reference to elements used for determination of conditional expressions.	Section 4.3.1.8
9.	DATA	Describes data.	Section 4.3.1.9
10.	DATATYPE	Describes data types.	Section 4.3.1.10
11.	DEFAULT	Describes default values.	Section 4.3.1.11
12.	END_CONDITION	Describes normal end conditions.	Section 4.3.1.12
13.	ENG_UNIT	Describes the engineering unit.	Section 4.3.1.13
14.	ERR_CODE_RANGE	Described the error code range.	Section 4.3.1.14
15.	ERR_CONDITION	Describes abnormal end conditions.	Section 4.3.1.15
16.	ERR_REGISTER	Describes the name of the register that stores error codes.	Section 4.3.1.16
17.	ERR_TYPE	Describes the type of the response processing data format when an error occurs.	Section 4.3.1.17
18.	INTERLOCK	Describes conditional expressions using register names.	Section 4.3.1.18
19.	LABEL	Describes labels.	Section 4.3.1.19
20.	LABEL2	Describes labels used for other language support.	Section 4.3.1.20
21.	MESSAGE_TYPE	Describes the message type.	Section 4.3.1.21
22.	METHOD_TYPE	Describes the method type.	Section 4.3.1.22
23.	MIN_INC	Describes the minimum increment.	Section 4.3.1.23
24.	NAME	Describes the name.	Section 4.3.1.24
25.	OFFSET	Describes the assigned offset of the structure interior.	Section 4.3.1.25
26.	RANGE	Describes the setting range.	Section 4.3.1.26
27.	READ_DATA	Describes the storage destination of read data.	Section 4.3.1.27
28.	READ_DATATYPE	Describes the data type of the register to be read.	Section 4.3.1.28
29.	READ_REGISTER	Describes the register that is to implement reading.	Section 4.3.1.29
30.	REF	Describes a reference.	Section 4.3.1.30
31.	RELATED_ELE	Describes a reference.	Section 4.3.1.31
32.	RELATED_MESSAGE	Describes a reference to a message that indicates processing prior to the message.	Section 4.3.1.32
33.	RELATED_METHOD	Describes a reference to a method that indicates processing prior to the method.	Section 4.3.1.33
34.	REQ_FLAG	Describes the request flag name.	Section 4.3.1.34
35.	REQUEST_DATA	Describes the storage destination of the value to be delivered in request processing.	Section 4.3.1.35
36.	REQUEST_DATATYPE	Describes the data type of REQUEST_DATA.	Section 4.3.1.36
37.	REQUEST_TYPE	Describes the type of the data format for request processing.	Section 4.3.1.37
38.	RESPONSE_DATA	Describes the value to be returned in response processing.	Section 0
39.	RESPONSE_DATATYPE	Describes the data type of RESPONSE_DATA.	Section 4.3.1.39
40.	RESPONSE_TYPE	Describes the type of the data format of the data used in response processing.	Section 4.3.1.40
41.	RESULTx	Writes the value applied when conditions are satisfied.	Section 0
42.	TARGET	Describes the communication parameters or communication commands to serve as the target of the method or message.	Section 4.3.1.42
43.	UI_ATTRIBUTE	Describes the display method when displaying data using utility software.	Section 4.3.1.43
44.	WRITE_DATA	Describes the write value.	Section 4.3.1.44
45.	WRITE_DATATYPE	Describes the data type of the register for writing.	Section 4.3.1.45
46.	WRITE_ORDER	Describes the writing order.	Section 4.3.1.46
47.	WRITE_REGISTER	Describes the register that is to implement writing.	Section 4.3.1.47

4.2.4 CSP+ file structure of CC-Link connected module

Figure 4-5 shows a general model of a CSP+ file of a CC-Link connected module.

The details of each part described in the COMM_IF section and BLOCK section as well as the reference relationships are as follows.

(1) COMM_IF section output and BLOCK section input
Output as viewed from the communication function side (same as output as viewed from the master station) is described in the COMM_IF_OUTPUT part of the COMM_IF section, and in the BLOCK_INPUT part of the BLOCK section.

The direction of input/output of the BLOCK section is the same as the input/output from the control function side to the communication function side. As a result, the output from the communication function side is the input to the control function side. Thus, the BLOCK_INPUT part corresponds to the COMM_IF_OUTPUT part.

The remote output (RY) and remote register (RWw) information is described in each element of the COMM_IF_OUTPUT part, and the module output contact and output area information as well as module set values, flags, and the like are described in each element of the BLOCK_INPUT part.

There is no need to describe data that is not mounted or used by the module. However, the addresses of remote outputs and remote registers for which corresponding elements are not described in cyclic transmission areas that can be used based on module settings indicate that the areas are not accessible by the sequence program and utility software. If you want to clearly indicate the reason why an area is not accessible, describe the non-accessible element with the reason written in the name.

Each element of the COMM_IF_OUTPUT part refers to the elements of the associated BLOCK_INPU part.

Non-associated elements may be present.

(2) COMM_IF section input and BLOCK section output
Input as viewed from the communication function side (same as input as viewed from the master station) is described in the COMM_IF_INPUT part of the COMM_IF section, and in the BLOCK_OUTPUT part of the BLOCK section.

The input to the communication function side is the output from the control function side. Thus, the BLOCK_OUTPUT part corresponds to the COMM_IF_INPUT part.

The remote input (RX) and remote register (RWr) information is described in each element of the COMM_IF_INPUT part, and the module input contact and input area information as well as module present values, status information, and the like are described in each element of the BLOCK_OUTPUT part.

There is no need to describe data that is not mounted or used by the module. However, the addresses of remote inputs and remote registers for which corresponding elements are not described in cyclic transmission areas that can be used based on module settings indicate that the areas are not accessible by the sequence program and utility software. If you want to clearly indicate the reason why an area is not accessible, describe the non-accessible element with the reason written in the name.

Each element of the COMM_IF_INPUT part refers to the elements of the associated BLOCK_OUTPUT part.

Non-associated elements may be present.

(3) Parameters
Parameters are described in the COMM_IF_PARAMETER part of the COMM_IF section, and the BLOCK_PARAMETER part of the BLOCK section.

The procedures for reading and writing parameters and information regarding the parameters used are described in the METHOD part.

Each element of the COM_IF_PARAMETER part and BLOCK_PARAMETER part is associated so that the elements of the METHOD part refer to the elements of the COMM_IF_PARAMETER part, and the elements of the COMM_IF_PARAMETER part refer to the elements of the BLOCK_PARAMETER part.

(4) Commands
Commands are described in the COMM_IF_COMMAND part of the COMM_IF section, and the BLOCK_COMMAND part of the BLOCK section.
The procedures for executing commands and information regarding the arguments used are described in the METHOD part.
Each element of the COMM_IF_COMMAND part and BLOCK_COMMAND part is associated so that the elements of the METHOD part refer to the elements of the COMM_IF_COMMAND part, and the elements of the COMM_IF_COMMAND part refer to the elements of the BLOCK_COMMAND part.

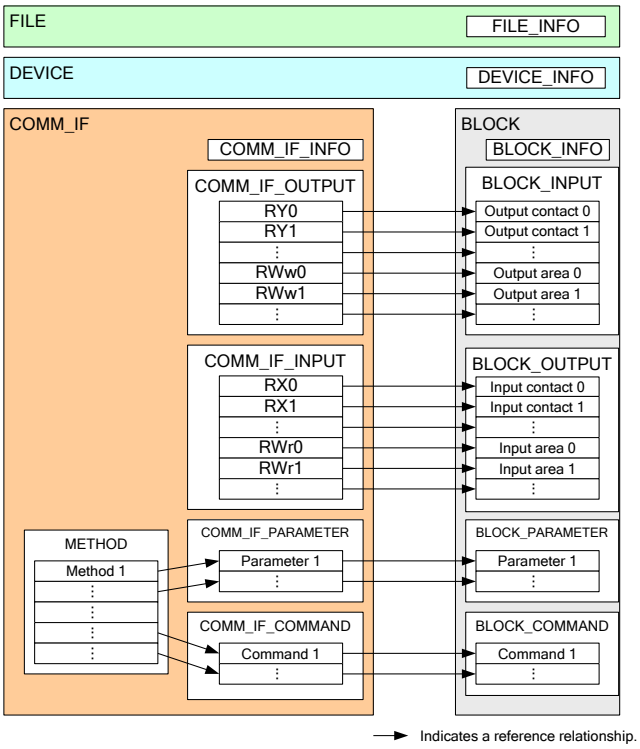


Figure 4-5 General Model of CSP+ File for CC-Link Connected Module

4.2.4.1 CSP+ file structure of remote I/O station

Figure 4-6 shows a model of the structure of a CSP+ file for a remote I/O station of a CC-Link connected module. With a remote I/O station, the parts COMM_IF_INPUT, COMM_IF_OUTPUT, BLOCK_INPUT, and BLOCK_OUTPUT are used.
With a remote I/O station, the parts METHOD, COMM_IF_PARAMETER, COMM_IF_COMMAND, BLOCK_PARAMETER, and BLOCK_COMMAND are not written. Further, for input/output, only remote input/output (RX, RY) is described.

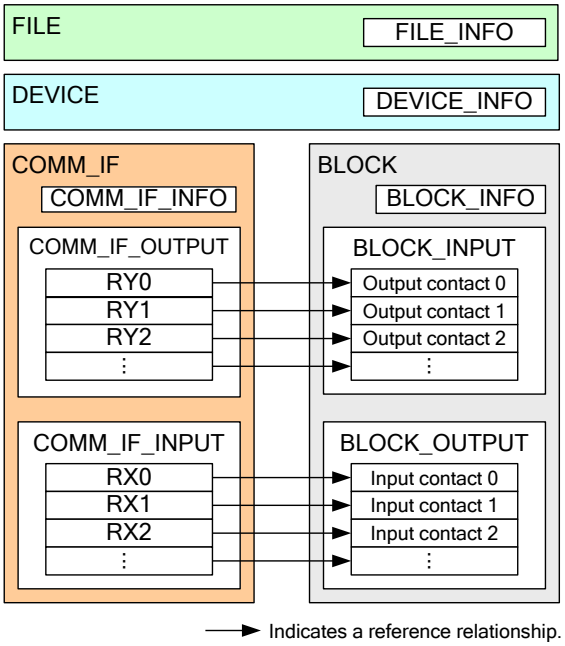


Figure 4-6 Model of CSP+ File for CC-Link Remote I/O Station

All remote I/O stations cannot necessarily be mounted with an input/output function. The mountable information is defined for each I/O type as shown in Figure 4-8.

Table 4-8 Definitions of Modules by I/O Type

No.	I/O Type	COMM_IF_OUTPUT Part, BLOCK_INPUT Part	COMM_IF_INPUT Part, BLOCK_OUTPUT Part
1.	Input	×	○
2.	Output	○	×
3.	Front/Back mixed, mixed	○	○

4.2.4.2 CSP+ file structure of remote device station, intelligent device station, and local station

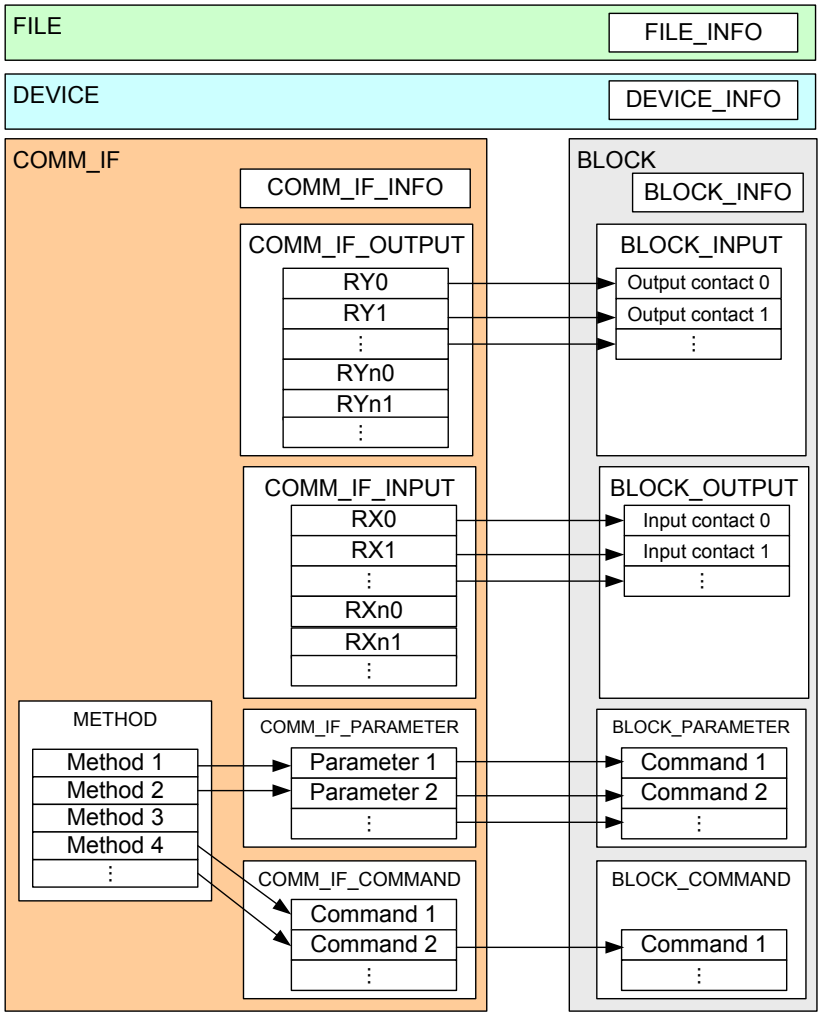
The models of a remote device station, intelligent device station, and local station of CC-Link connected module are the same as that of a general model. Figure 4-7 shows a model of the CSP+ file.
Note that there is no need to describe references from all elements of the COMM_IF section to the BLOCK session. An example is described below.

[1] CC-Link system area

[2] Communication module dedicated parameter and command information of communication module type modules

[3] Input/Output information not directly related with the device control function, such as flags for method handshakes

While a local station is capable of referring to values outputted by all stations in the same manner as a master station, only the input/output information assigned to the location station (the input/output information that can be read/written by the local station only) is described in the COMM_IF_OUTPUT part and COMM_IF_INPUT part. At this time, the last two points of remote input/output (RX, RY) are not described. As a result, for example, the remote input/output of 30 points is described for one occupied station at 1x, and the remote input/output of 446 points is described for four occupied stations at 4x.



→ Indicates a reference relationship.

n: Indicates the system area. (No. of occupied stations: ·2-1)
*However, if the device if fixed to 32 I/O points, n = 1, regardless of the number of occupied stations.

Figure 4-7 Model of CSP+ File for CC-Link Remote Device Station, Intelligent Device Station, and Local Station

4.2.4.3 CSP+ file structure when multiple COMM_IF sections are written

When the number of occupied stations (1, 2, 3, or 4), the CC-Link version (1.00, 1.10, 1.11, 2.00), or the extended cyclic setting (with a CC-Link version of 2.00) is variable, a COMM_IF section needs to be written for each setting. (The section needs to be described taking combinations into consideration as well.) With the utility software, only the settings described in COMM_IF sections need to be acknowledged.

If the communication specifications change as a result of changes in module operation settings, regardless if the above settings remain the same, we recommend that you write a COMM_IF section for each operation setting. In such a case, if multiple COMM_IF sections are written, the content indicated by each COMM_IF section needs to be described in the COMMENT item.

Note that, if the module function does not change even if multiple COMM_IF sections are written, the associations from the multiple COMM_IF sections to a common BLOCK section are to be described.

Figure 4-8 shows the section configuration in a case where the number of occupied stations is selectable from the values 1 to 4.

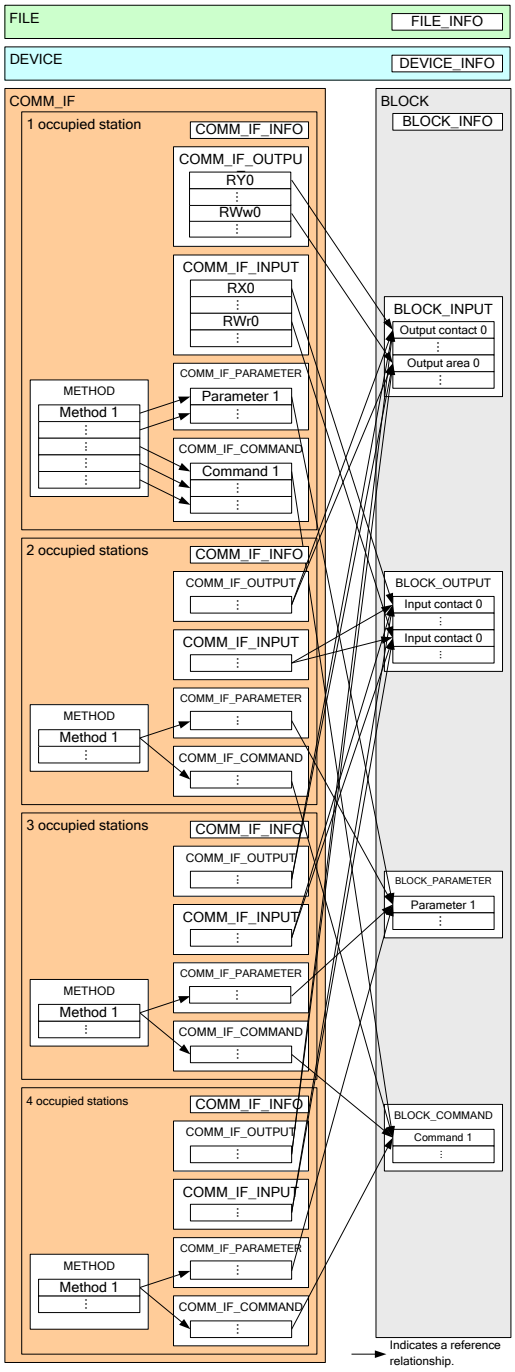


Figure 4-8 Description Example of Multiple COMM_IF Sections with CC-Link

4.2.5 CSP+ file structure of CC-Link IE Field Network connected modules

Figure 4-9 shows a general model of a CSP+ file of a CC-Link IE Field Network connected module. With a CC-Link IE Field Network connected module, you can write messages for accessing inputs, outputs, parameters, and commands using SLMP in addition to parameter setup and command execution using the same methods as with CC-Link connected modules. The differences in the reference relationship with the contents of each part described in the COMM_IF and BLOCK sections from that with a CC-Link connected module (see Section 4.2.4) are as follows.

(1) COMM_IF section output and BLOCK section input
In addition to the contents that can be written with CC-Link connected modules, the following descriptions are also possible.
Information regarding the input contacts and input areas of data read using SLMP commands is described in each element of the COMM_IF_INPUT part. In addition, information related to SLMP commands used for data reading is described in the MESSAGE part of the COMM_IF section. The descriptions of each element of the BLOCK_OUTPUT part are the same as that for access via cyclic transmission. Each element of the COMM_IF_INPUT part is associated by reference from with the elements of the MESSAGE part, with the elements of the MESSAGE part referring to the elements of the COMM_IF_INPUT part. (The reference relationship between the elements of the COMM_IF_INPUT part and each element of the BLOCK_OUTPUT part is the same as that for access via cyclic transmission.)
Note that access via SLMP and access via cyclic transmission can be used in combination. In such a case, as shown in Figure 4-9, the BLOCK_OUTPUT part is referred to by two COMM_IF_INPUT parts.

(2) COMM_IF section input and BLOCK section output
In addition to the contents that can be written with CC-Link connected modules, the following descriptions are also possible.
Information regarding the output contacts and output areas of data to be written using SLMP commands is described in each element of the COMM_IF_OUTPUT part. In addition, information related to SLMP commands to be used for data writing is described in the MESSAGE part of the COMM_IF section. The descriptions of each element of the BLOCK_INPUT part are the same as those for access via cyclic transmission. Each element of the COMM_IF_OUTPUT part is associated by reference with the elements of the MESSAGE part, with the elements of the MESSAGE part referring to the elements of the COMM_IF_OUTPUT part. (The reference relationship between the elements of the COMM_IF_OUTPUT part and each element of the BLOCK_INPUT part is the same as that for access via cyclic transmission.)

Note that access via SLMP and access via cyclic transmission can be used in combination. In such a case, as shown in Figure 4-9, the BLOCK_INPUT part is referred to by two COMM_IF_OUTPUT parts.

(3) Parameters
In addition to the contents that can be written with CC-Link connected modules, the following descriptions are also possible.
Information related to SLMP commands to be used for parameter reading and writing is described in the MESSAGE part of the COMM_IF section. Each element of the COMM_IF_PARAMETER part is associated by reference with the elements of the MESSAGE part, with the elements of the MESSAGE part referring to the elements of the COMM_IF_PARAMETER part. (The reference relationship between the elements of the COMM_IF_PARAMETER part and the elements of the BLOCK_PARAMETER part is the same as that for access via cyclic transmission.)

(4) Commands
In addition to the contents that can be written with CC-Link connected modules, the following descriptions are also possible.

Information related to SLMP commands to be used for command execution is described in the MESSAGE part of the COMM_IF section. Each element of the COMM_IF_COMMAND part is associated by reference with the elements of the MESSAGE part, with the elements of the MESSAGE part referring to the elements of the COMM_IF_COMMAND part. (The reference relationship between the elements of the COMM_IF_COMMAND part and the elements of the BLOCK_COMMAND part is the same as that for access via cyclic transmission.)

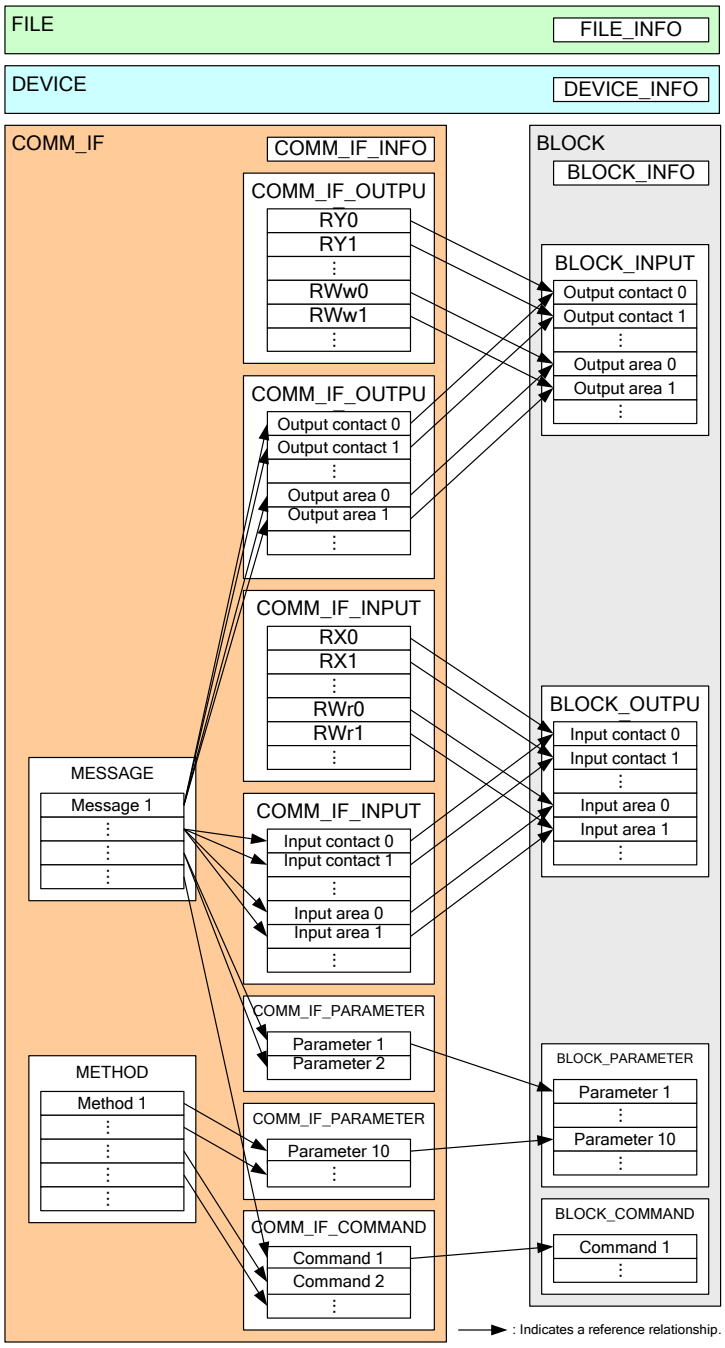


Figure 4-9 General Model of CSP+ File of CC-Link IF Field Network Connected Module

4.2.5.1 Structure of CSP+ file of remote I/O station

Figure 4-10 shows a model of the structure of a CSP+ file of a remote I/O station of a CC-Link IE Field Network connected module. With a remote I/O station, the parts COMM_IF_INPUT, COMM_IF_OUTPUT, BLOCK_INPUT, and BLOCK_OUTPUT are used.
The parts METHOD, MESSAGE, COMM_IF_PARAMETER, COMM_IF_COMMAND, BLOCK_PARAMETER, and BLOCK_COMMAND cannot be written with a remote I/O station. Further, for input/output as well, only remote input/output (RX, RY) is described.

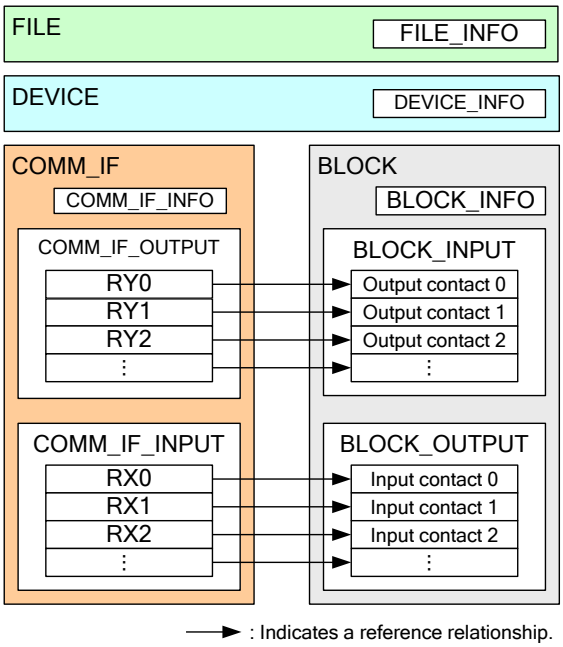


Figure 4-10 Model of CSP+ File for CC-Link IE Field Network Connected Remote I/O

All remote I/O stations are not necessarily capable of mounting an input/output function. As shown in Table 4-8, the mountable information is defined for each I/O type.

4.2.5.2 CSP+ file structure of remote device station, intelligent device station, and local station

The model for a remote device station, intelligent device station, and local station of a CC-Link IE Field Network connected module is the same as the general model. Figure 4-11 shows a model of the CSP+ file. Note that there is no need to describe references from all elements of the COMM_IF section to the BLOCK session. Examples are described below.

(1) System area of CC-Link IE Field Network

(2) Communication module dedicated parameter and command information of communication module type module

For the parts COMM_IF_INPUT and COMM_IF_OUTPUT, the data that is refreshed in the cyclic area and the data that is accessed by SLMP (referred to by a MESSAGE part) need to be described in different parts.
Further, while remote inputs/outputs that cannot be described exist for a CC-Link connected local station, all remote inputs/outputs can be described for a CC-Link IE Field Network connected local station.
(The fact that only the input/output information assigned to a local station is described in the parts COMM_IF_OUTPUT and COMM_IF_INPUT holds true for CC-Link connected local stations as well.)

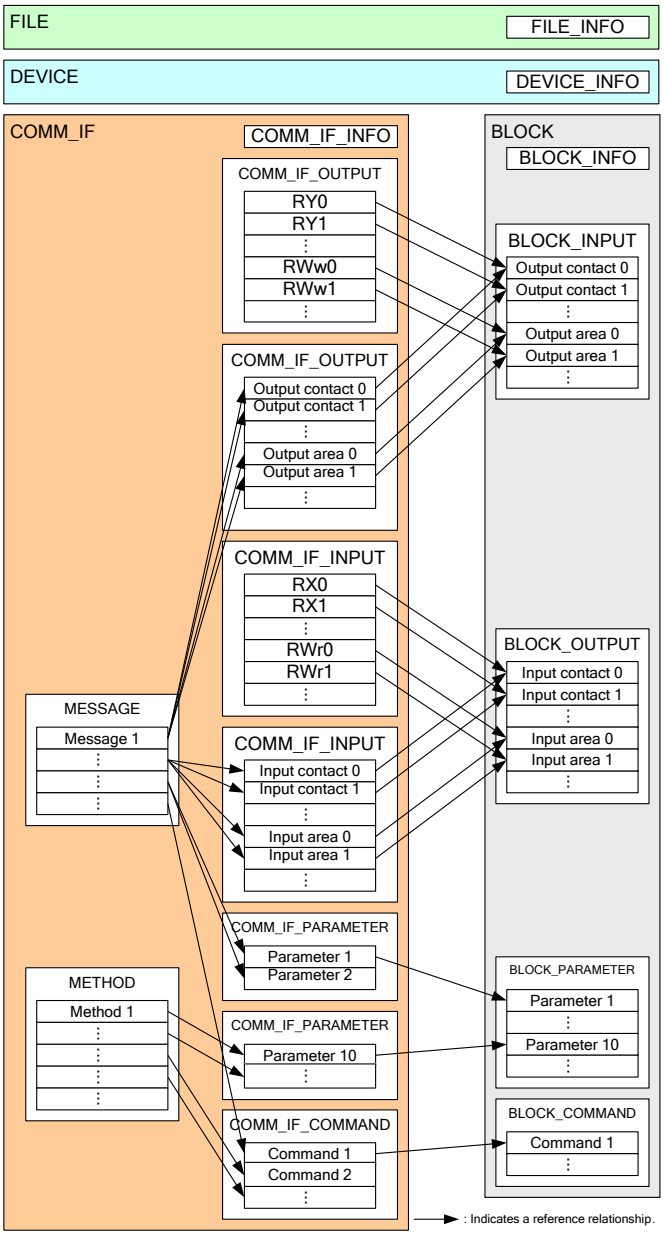


Figure 4-11 Model of CSP+ File of CC-Link IE Field Network Connected Remote Device Station, Intelligent Device Station, and Local Station

4.2.5.3 CSP+ file structure when multiple COMM_IF sections are written

If communication specifications change when module operation settings are changed, we recommend that you write a COMM_IF section for each operation setting. If, in this case, multiple COMM_IF sections are written, an explanation of what is indicated by each COMM_IF section needs to be written in COMMENT.

Such a scenario applies to a module in which the operation mode can be set to CC-Link compatible mode (in which SLMP cannot be used), for example. In this example, as shown in Figure 4-12, the file describes a COMM_IF section in which SLMP is applicable, and a COMM_IF section in which SLMP is not applicable (a COMM_IF section for CC-Link compatible mode).

To write multiple COMM_IF sections in CC-Link IE TSN, use conventions described in Section 4.2.8.2.

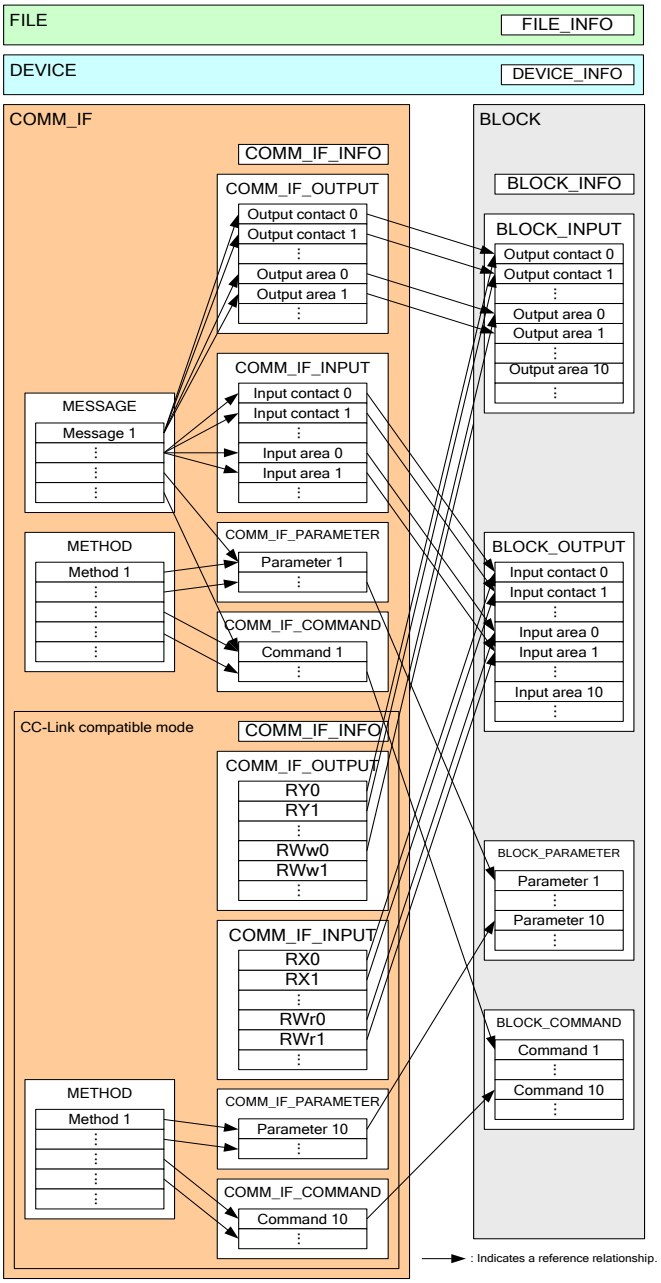


Figure 4-12 Writing Multiple COMM_IF Sections in a CC-Link IE Field Network

4.2.6 In the case of the CC-Link IE Field Network Safety Communication Function Support Unit

In the case of the CC-Link IE Field Network Safety Communication Function Support Unit, it is necessary to write specific information such as special device names and elements. In the case of the Safety Communication Function Support Unit, in addition to the description of the unit of the normal Field Network, the following descriptions are also conducted.

(1) COMM_IF_INFO Part safety communication function support element (See Section 5.3.1)

(2) CONDITION Part (See Section 4.3.1.7)

(3) Safety communication function parameters (See Section 5.5.5)

(4) Messages for safety parameter writing (See Section 5.5.6)

4.2.7 CSP+ file structure of Ethernet connection unit

Figure 4-13 shows a general model of a CSP+ file of a unit connected to the Ethernet.
For the Ethernet connection unit, parameters can be described for reading and writing parameters of the unit by specifying parameter IDs. When doing so, since specific SLMP commands are prepared and used to perform communication, messages are not described.

(1) COMM_IF section output and BLOCK section input
Output as viewed from the communication function side (same as output as viewed from the master station) is described in the COMM_IF_OUTPUT part of the COMM_IF section, and in the BLOCK_INPUT part of the BLOCK section.
The direction of the input/output of the BLOCK section is the same as the input/output from the control function side to the communication function side. As a result, the output from the communication function is the same as the input to the control function side. Thus, the BLOCK_INPUT part corresponds to the COMM_IF_OUTPUT part.
The remote output (RY) and remote register (RWw) information is described in each element of the COMM_IF_OUTPUT part, and the module output contact and output area information as well as module set values, flags, and the like are described in each element of the BLOCK_INPUT part.
There is no need to describe data that is not mounted or used by the module. However, the addresses of remote outputs and remote registers for which corresponding elements are not described in cyclic transmission areas that can be used based on module settings indicate that the areas are not accessible by the sequence program and utility software. If you want to clearly indicate the reason why an area is not accessible, describe the non-accessible element with the reason written in the name.
Each element of the COMM_IF_OUTPUT part refers to the elements of the associated BLOCK_INPUT part.
Non-associated elements may be present.

(2) COMM_IF section input and BLOCK section output
Input as viewed from the communication function side (same as input as viewed from the master station) is described in the COMM_IF_INPUT part of the COMM_IF section, and in the BLOCK_OUTPUT part of the BLOCK section.
The input to the communication function side is the output from the control function side. Thus, the BLOCK_OUTPUT part corresponds to the COMM_IF_INPUT part.
The remote input (RX) and remote register (RWr) information is described in each element of the COMM_IF_INPUT part, and the module input contact and input area information as well as module present values, status information, and the like are described in each element of the BLOCK_OUTPUT part.
There is no need to describe data that is not mounted or used by the module. However, the addresses of remote inputs and remote registers for which corresponding elements are not described in cyclic transmission areas that can be used based on module settings indicate that the areas are not accessible by the sequence program and utility software. If you want to clearly indicate the reason why an area is not accessible, describe the non-accessible element with the reason written in the name.
Each element of the COMM_IF_INPUT part refers to the elements of the associated BLOCK_OUTPUT part.
Non-associated elements may be present.

(3) Parameter
Parameters that perform read/write can be written by specifying parameter ID.
Parameters are written in association by writing them as elements of COMM_IF_PARAMETER part and BLOCK_PARAMETER part, and referring elements of BLOCK_PARAMETER part from elements of COMM_IF_PARAMETER part. (See Section 5.5.7.)

(4) Command
Commands are described in the COMM_IF_COMMAND part of the COMM_IF section, and the BLOCK_COMMAND part of the BLOCK section.
The elements of the METHOD part refer to the elements of the COMM_IF_COMMAND part, and the elements of the COMM_IF_COMMAND part refer to the elements of the BLOCK_COMMAND part. Thus, each element of the COMM_IF_COMMAND part and BLOCK_COMMAND part is associated.

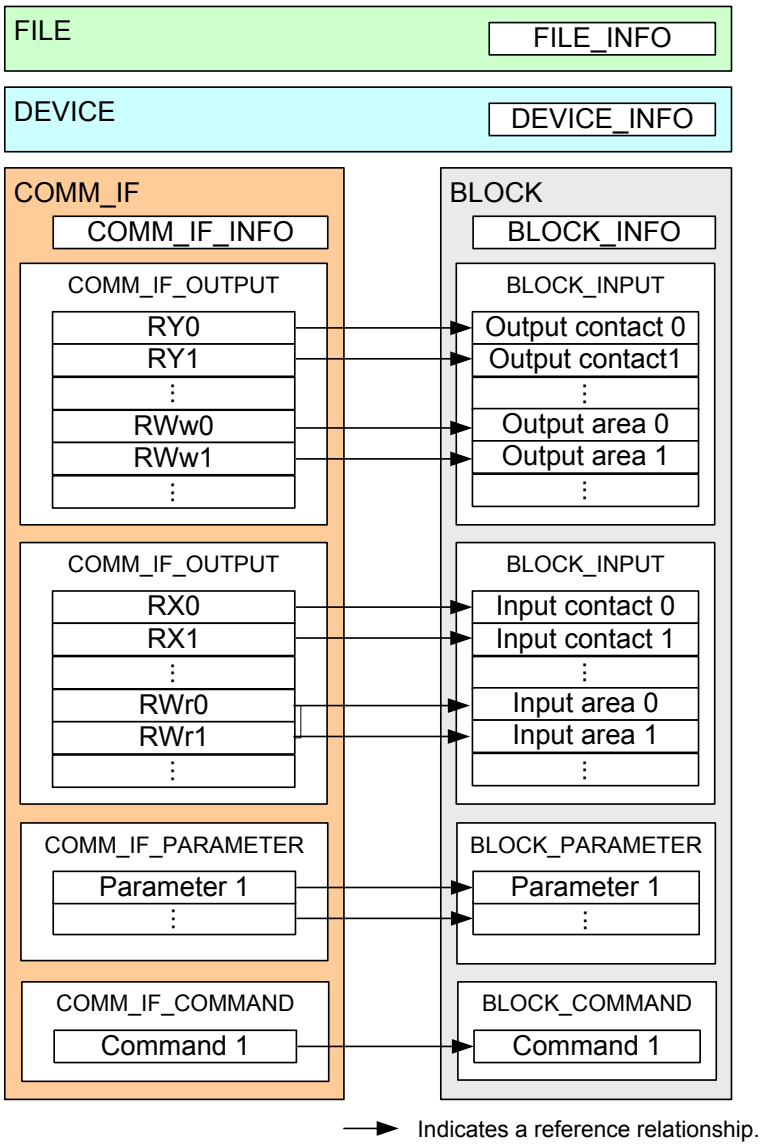


Figure 4-13 General Model of CSP+ File for Ethernet Connection Unit

* At the time of version 2.1, only one type of command (status code acquisition command) is defined.

4.2.7.1 CSP+ file structure when multiple COMM_IF sections are written

When the number of occupied stations (1 to 64) is variable, a COMM_IF section needs to be written for each setting. With the utility software, only the settings described in COMM_IF sections need to be acknowledged.

If the communication specifications change as a result of changes in module operation settings, regardless if the above settings remain the same, we recommend that you write a COMM_IF section for each operation setting. In such a case, if multiple COMM_IF sections are written, the content indicated by each COMM_IF section needs to be described in the COMMENT item.

Note that, if the module function does not change even if multiple COMM_IF sections are written, the associations from the multiple COMM_IF sections to a common BLOCK section are to be described.

Figure 4-14 shows the section configuration in a case where the number of occupied stations is selectable from the values 1 to 4.

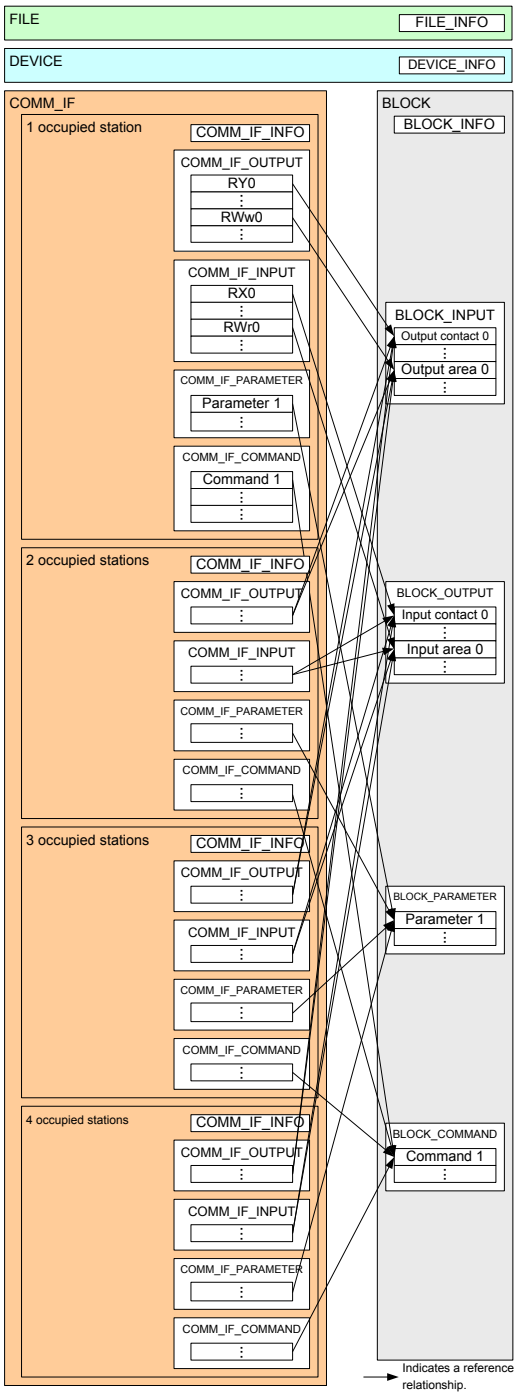


Figure 4-14 Description Example of Multiple COMM_IF Sections with Ethernet Connection Unit

4.2.8 CSP+ file structure of the CC-Link IE TSN connection unit

Figure 4-15 shows a general model of a CSP+ file of the CC-Link IE TSN connection unit.
With a CC-Link IE TSN connection unit, you can write messages for accessing inputs, outputs, parameters, and commands using parameter setup, command execution, and SLMP using the same methods as with the CC-Link IE Field Network connection unit. There are no differences in the reference relationship with the contents of each part described in the COMM_IF and BLOCK sections from that with a CC-Link IE Field Network connected module (see Section 4.2.5). In the case of CAN compatible devices, the object dictionary information needs to be written and added as a CSV file. (For details on creating an object dictionary file, see Section 4.2.8.4.)

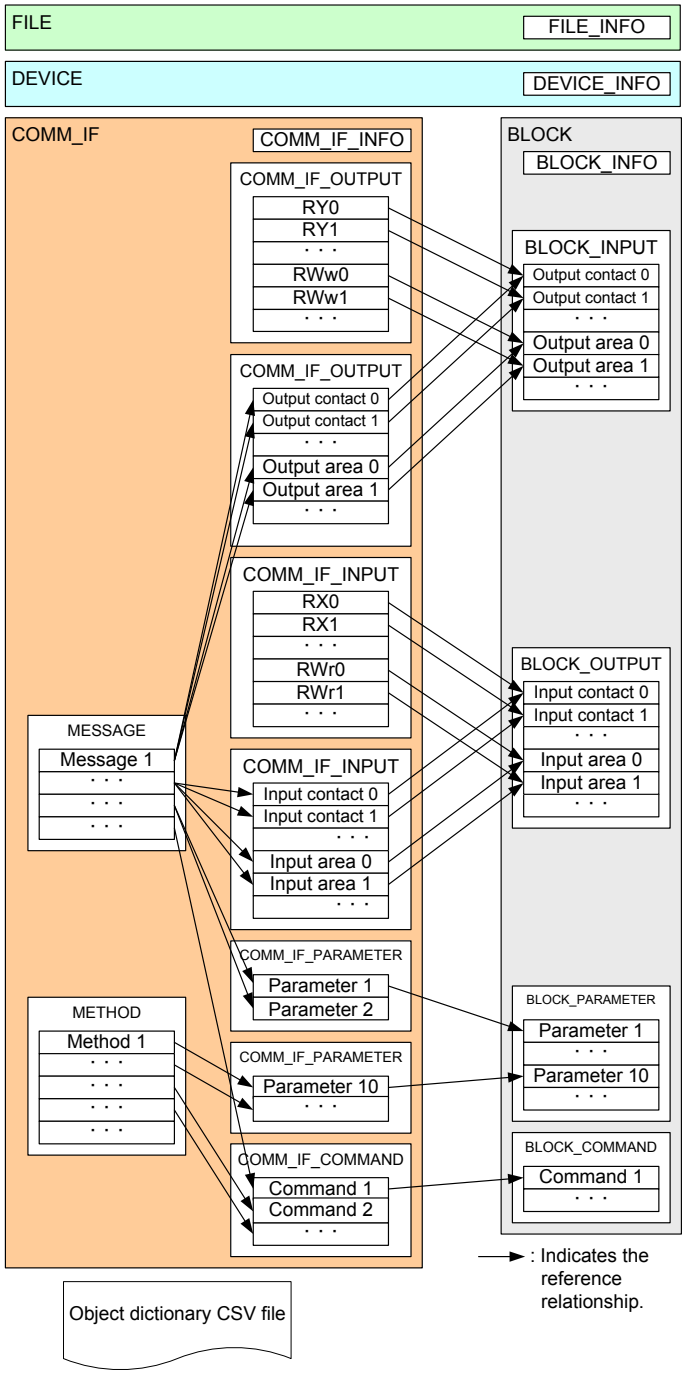


Figure 4-15 General Model of CSP+ File of CC-Link IE TSN Connection Unit

4.2.8.1 CSP+ file structure of a local station (slave station with the local station function enabled)

The model of a local station (slave station with the local station function enabled) of a CC-Link IE TSN connection unit is the same as that of a general model. (See Figure 4-15.)
With regard to the descriptions of CC-Link IE TSN connected local stations, a local station is capable of acquiring the values output by all stations in the same manner as a master station, but only the input/output information assigned to the location station is written in the COMM_IF_INPUT and COMM_IF_OUTPUT parts.

4.2.8.2 Devices for which station mode can be changed

In the case of devices for which station mode can be changed, set different COMM_IF sections for every operation setting **[Required]**. (For details on the station mode, see CC-Link IE TSN Specifications (Overview) BAP-C2011ENG-001.)

- Information on the station mode is written in the StationMode element (Section 5.3.1.3(53)) of the COMM_IF_INFO part. **[Required]** Figure 4-16 shows a description image of devices for which each mode can be switched.
- If the device specifications do not change even though the information of station mode is changed, create one common BLOCK section and set the BLOCK section to be referred from multiple COMM_IF sections.
- When multiple COMM_IF sections are set and the device can acquire the error related information, matching descriptions of each COMM_IF section is required. For details, see Section 5.5.10.1.

For switching the mode with other than CC-Link IE TSN, use the conventions described in Section 4.2.5.3.

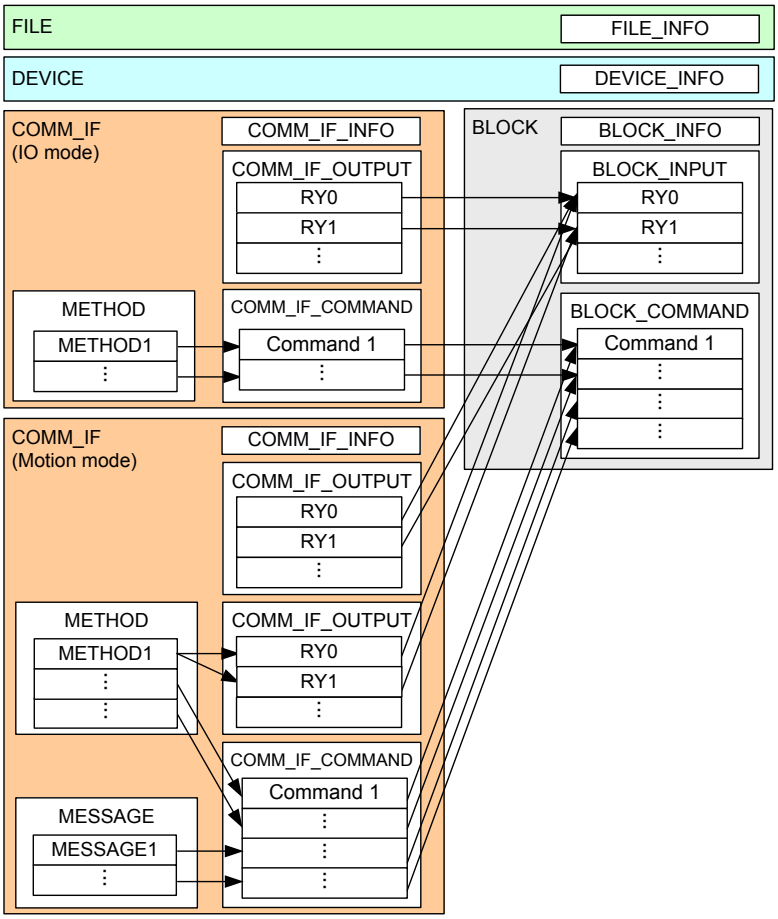


Figure 4-16 Example of CC-Link IE TSN Devices for which Station Mode Can Be Changed

4.2.8.3 Description specifications in common between devices

In the case of devices that can execute the functions described in this section of CC-Link IE TSN, the COMM_IF_COMMAND part needs to follow the description method using MESSAGE.

[1] Acquiring error related information
In the case of devices that can acquire error codes and error history information using SNMP, use the MESSAGE part for the diagnostics code information for description. For details, see Section 5.5.10.1.

[2] MESSAGE for slave station parameter automatic setting
This function automatically writes the parameters when a CC-Link IE TSN slave station is reconnected or introduced by turning on its power. Use MESSAGE for slave station parameter automatic setting for description. For details, see Section 5.5.10.2.

4.2.8.4 Object dictionary description specifications

(1) Overview
Various data such as control parameters and instruction values held by a CAN compatible device of a CC-Link IE TSN connection unit are treated as objects composed of Index, SubIndex, and others of the description details. A collection of these objects is an object dictionary.

(2) CSV file description specifications
Describe the object dictionary in a CSV file which is different from the CSP+ file. For details on the configuration in the compressed file, see Section 4.1.2.

For the description method and details of the CSV file, see Chapter 8.

4.2.9 In the case of the CC-Link IE TSN safety communication function support unit

In the case of the CC-Link IE TSN safety communication function support unit, it is necessary to write specific information such as special device names and elements. For a safety communication function support unit, describe the following in addition to the descriptions of a normal CC-Link IE TSN unit.

(1) CC-Link IE TSN safety communication function compatible elements of COMM_IF_INFO part (see Section 5.3.1)

(2) Parameters of safety communication function (see Section 5.5.10.3)

(3) Messages for writing safety parameters (see Section 5.5.10.4)

4.2.10 Structure of sections, parts, and elements of a CSP+ file

Figure 4-17 to Figure 4-20 show images of the structures of each section written in a CSP+ file, in XML format.

```
Structure of FILE section
<p:file label="FileSection">
    Structure of part
    <p:fileInfo label="FileInfo">
        Structure of element
        <p:fileInfoMember label="CreateDate">
            <p:label2><p:item> File creation date </p:item></p:label2>
            <p:category><p:item>COMMON</p:item></p:category>
            <p:name><p:item> File creation date </p:item></p:name>
            <p:datatype><p:item>STRING_FLEX(10)</p:item></p:datatype>
            <p:data><p:item>2011/12/01</p:item></p:data>
        </p:fileInfoMember>
        :
        </p:fileInfo>
</p:file>
```

Figure 4-17 Structure of FILE Section

```
Structure of DEVICE section
<p:device label="DeviceSection">
    Structure of part
    <p:deviceInfo label="DeviceInfo">
        Structure of element
        <p:deviceInfoMember label="VendorName">
            <p:label2><p:item> Vendor name </p:item></p:label2>
            <p:category><p:item>COMMON</p:item></p:category>
            <p:name><p:item> Vendor name </p:item></p:name>
            <p:datatype><p:item>STRING_FLEX(64)</p:item></p:datatype>
            <p:data><p:item> CC-Link Association </p:item></p:data>
        </p:deviceInfoMember>
        :
        </p:deviceInfo>
</p:device>
```

Figure 4-18 Structure of DEVICE Section

Structure of COMM_IF section

```
<p:commIf label="CommIfSection">
  Structure of part
    <p:commIfOutput label="CommIfOutput">
      Structure of element
        <p:commIfInputMember label="CH1_digital_output_val">
          <p:label2><p:item>CH1_ Digital output value </p:item></p:label2>
          <p:name><p:item>CH1 Digital output value </p:item></p:name>
          <p:datatype><p:item>INT16</p:item></p:datatype>
          <p:access><p:item>RF_IN</p:item></p:access>
          <p:assign><p:item>RWr0</p:item></p:assign>
          <p:ref>
          <p:item>BlockSection.BlockOutput.CH1_digital_output_val</p:item>
          </p:ref>
          <p:comment>
          <p:item> Enable if remote READY and CH1 A/D conversion completed flag are ON </p:item>
          </p:comment>
        </p:commIfInputMember>
    :
    </p: commIfOutput>
:
</p:commIf>
```

Figure 4-19 Structure of COMM_IF Section

Structure of BLOCK section

```
<p:block label="BlockSection">
  Structure of part
    <p:blockOutput label="BlockOutput">
      Structure of element
        <p:blockOutputMember label="CH1_digital_output_val">
          <p:label2><p:item>CH1_ Digital output value </p:item></p:label2>
          <p:name><p:item>CH1 Digital output value </p:item></p:name>
          <p:datatype><p:item>INT16</p:item></p:datatype>
          <p:default><p:item>0</p:item></p:default>
          <p:access><p:item>RF_OUT</p:item></p:access>
          <p:comment>
          <p:item> Enable if remote READY and CH1 A/D conversion completed flag are ON </p:item>
          </p:comment>
        </p:blockOutputMember>
    :
    </p:blockOutput>
:
</p:block>
```

Figure 4-20 Structure of BLOCK Section

4.3 SCP+ File Description Specifications

The following describes the conventions that are common to each section.

4.3.1 Item conventions

This section describes item conventions.

(1) Describe all items other than LABEL as an "XML element." Describe a LABEL item as an "attribute of an XML element."
(2) Describe a LABEL item as information that identifies sections and parts other than elements.
(3) Describe a COMMENT item as comment information that identifies sections and parts other than elements.
(4) If an item has an arbitrary description and does not require value setup, do not describe anything following the item tags.
(5) If you want to describe a reference to a STRUCT part, describe the reference as an element of "structRefItem." The following shows an example of a case where reference to a STRUCT part is described in a DATATYPE item.
(Example: <p:datatype><p:structRefItem>STMonitorCode1and2</p:structRefItem></p:datatype>)
The items DEFAULT, RANGE, MIN_INC, ENG_UNIT, and REF cannot be written in an element of a structure type (in an element where the DATATYPE is STRUCT).
(6) If you want to describe a reference to an ENUM part, describe the reference as an element of "enumRefItem." The following shows an example of a case where reference to a ENUM part is described in a RANGE item.
(Example: <p:range><p:enumRefItem>enumFaultDefinition</p:enumRefItem></p:range>)
(7) If you want to describe a reference to a RESULTx item of a CONDITION part, describe the reference as an element of "conditionRefItem". The following shows an example of a case where reference to a RESULT1 item of a CONDITION part is described in an ACCESS item.
(Example:
<p:access><p:conditionRefItem>C_AvgTimeCount.RESULT1</p:conditionRefItem></p:access>)
(8) Describe all items other than the above three cases and LABEL as an element of "item". The following shows an example of a case where a NAME item is described.
(Example: <name><item> Product name </item></name>)
(9) The 6 items LABEL2, NAME, CATEGORY, DATA, ENG_UNIT, and COMMENT require modification with support for other languages.

The following shows item description examples.

- Description example of element that uses "item" only
If common information such as STRUCT is not used, describe the element as shown below. (The following example is a case where the value of the item REF is "CH2.BlockParamter.AvgOrSample".)

```
<ref>
        <item>Ch2.BlockParamter.AvgOrSample</item>
</ref>
```

- Description example of item that uses structure information (STRUCT)
If the value of the item DATATYPE is structure information, describe structRefItem. The following shows a description example.

```
<datatype>
        <structRefItem>Prm_Struct</structRefItem>
</datatype>
```

- Description example of item that uses an option list (ENUM)
If the value of the item RANGE is an option list, describe enumRefItem. The following shows a description example.

```
<range>
        <enumRefItem>enumFaultDefinition</enumRefItem>
</range>
```

- Description example of a case where a set of values is described in an item
If you want to describe multiple values in a single item, describe the values by dividing the item into multiple items. The following shows a description example.

```
<data>
```

```
        <item>11:DC input</item>
        <item>12:DC input</item>
    </data>
```

4.3.1.1 ACCESS conventions

An ACCESS item describes access using the values shown in Table 4-9, according to part type.
For parameter and command argument elements, write "R," "W," or "RW," which indicates the reading/writing attribute.
Write "RF" for input and output elements capable of refresh by cyclic communication, etc.
Further, write "NA" for elements that are not accessible, regardless of the parameter or input/output, and omit the description if the element does not require access setup. If setup is omitted, access is in accordance with the access attribute of the element set as a reference. However, when setup is omitted in the parts BLOCK_INPUT and BLOCK_OUTPUT, the setup is regarded as writable (BLOCK_INPUT part) or readable (BLOCK_OUTPUT part).
ACCESS cannot be written in an element of a STRUCT part. The ACCESS value of an element of a STRUCT type of the reference source (an element where the DATATYPE is STRUCT) is applied to all elements within the STRUCT part. Further, even if ACCESS is written in an element referred to by a REF item from each element of the STRUCT part, that value is not used.

Table 4-9 ACCESS Attribute List

No.	Value	Meaning	Settable Part Type ○: Settable ×: Not settable			
			COMM_IF_ PARAMETER	BLOCK_PARAMETER COMMAND_ARGUMENT	COMM_IF_INPUT COMM_IF_OUTPUT	BLOCK_INPUT BLOCK_OUTPUT
1.	"R"	Readable	○	○	×	×
2.	"W"	Writable	○	○	×	×
3.	"RW"	Readable and writable	○	○	×	×
4.	"RF"	Auto refreshable	×	×	○	×
5.	"NA"	Element not accessible (Used with elements that are not accessible according to conditions, etc.)	○	○	○	○
6.	No item	No setting (Used when settings are determined according to the access attribute of the element of the reference)	○	×	○	○

4.3.1.2 ARGUMENT conventions

An ARGUMENT item describes label references for a COMMAND_ARGUMENT part (see Section 5.3.10) in order to indicate the arguments to be used by the element. For label reference conventions, see Section 4.3.4.

4.3.1.3 ASSIGN conventions

An ASSIGN item describes codes and addresses assigned to the element. This item may be omitted if not needed. For the method used to express the addresses for module specific resources (module internal memory, internal register, etc.),refer to the specifications of the module.
The remote input, remote output, and remote register addresses of CC-Link, the CC-Link IE Field Network, and CC-Link IE TSN are expressed as follows.

- Write the prefixes which indicate address type as follows:
  - Remote input: RX
  - Remote output: RY
  - Remote register (write area): RWw
  - Remote register (read area): RWr

  * The upper case and lower case letters are fixed as shown above. (Example: "Rx0" is not permitted.)
- The description of the address type is standardized to the type as viewed from the master module. For example, while the meanings of RX and RY of a local station are the reverse of the meanings

of a remote I/O in CC-Link specifications, write RX and RWr for the address (COMM_IF_INPUT) that is to send data to the master station, and RY and RWw for the address (COMM_IF_OUTPUT) that is to receive data from the master station.

- Write the address number as described below.
  - Add the number (hexadecimal) after the prefix (example: RX12, RYC, RWw1, RWrA).
  - While n, which indicates an address assigned to the master module by station number setup, is sometimes written with protocol specifications and the like, such as RXn9, n must not be written in profile descriptions.
  - If the initial value of the number is any value between A to F, you can simply add 0 or 1 before that number (example: RYC can be written as RY0C, and RXD2 can be written as RX0D2).
- When you want to specify a bit register within a word register (remote register), enter a dot (".") after the string indicating the word register, and write the bit register number in hexadecimal format (example: bit 7 of RWw3 is written as "RWw3.7").
- The letters A to F, which indicate hexadecimal numbers, can only be used in upper case. Lower case is not permitted.
- The size of the area used by the element is regulated by data type.
- Expressions that extend across multiple word devices cannot be written (example: a data type of UINT8, an ASSIGN item of W0.A). Note, however, that such an expression can be written if the element is assigned to each one of the multiple word devices (example: a data type of UINT32, and an ASSIGN item of W1 are acceptable).
- When the ASSIGN item is written in CC-Link IE TSN, the following restrictions apply.
  - Duplication of the ASSIGN items is prohibited among the elements in the same section.

    1) Duplication among all elements referenced from TARGET of MESSAGE for parameter writing is prohibited.
       - For a MESSAGE part where MESSAGE_TYPE item for MESSAGE for parameter writing is PARAMETER and reference to VALUE is written in REQUEST_DATA

    2) Duplication among all elements referenced from TARGET of MESSAGE for slave station parameter auto setting (see Section 5.5.10.2) is prohibited.

    The ASSIGN items can be duplicated among the above 1) and 2) elements. However, in that case, not only the ASSIGN items but also values of all items must be the same.

    [Example]

    When CommIfParameter1 part is referred in the TARGET item of MESSAGE for parameter writing and CommIfParameter2 part is referred in TARGET of MESSAGE for slave station parameter auto setting, the ASSIGN in the element of CommIfParameter1 and element of CommIfParameter2 can be duplicated. However, these are only different in writing method and the writing target is the same. When the NAME or others are different, the engineering tool users utilizing CSP+ may misunderstand, therefore, write same items to display the same in the parameter writing screen and parameter auto setting screen.

Further, an address number can also be used to specify a memory number during access via SLMP. In such a case, for the ASSIGN item, be sure to write an appropriate value that can be used by the REQUEST_DATA item since the ASSIGN item is referred to by the REQUEST_DATA item (see Section 4.3.1.35).

4.3.1.4 CATEGORY conventions

A CATEGORY item describes the group name with which the element is affiliated. For elements that you want to arrange in the same group, write the same group name in the CATEGORY item. If the element is not to be categorized, this item can be omitted.
When other languages are supported, write this label name in keeping with the descriptions of the LANGUAGE element and FILE_INFO part of the FILE section.

4.3.1.5 CODE conventions

A CODE item describes the value that identifies an element of the part ENUM (see Section 5.3.9.3). The CODE item also allows you to set a value that is out of range for the DATATYPE item (see Section 4.3.1.10) of an element that uses the part ENUM. As a result, the same ENUM part can be made common to elements of DATATYPE items having different value ranges.
Integers (decimal/hexadecimal) can be written in CODE items.

4.3.1.6 COMMENT conventions

A COMMENT item describes information such as the element itself, the meaning of each value, and usage precautionary points, using any string. Utility software uses the COMMENT item when displaying the data as additional information on the screen. This item may be omitted when not needed.
When other languages are supported, write this label name in keeping with the descriptions of the LANGUAGE element and FILE_INFO part of the FILE section.
The following describes the precautions to follow when writing a COMMENT.

(1) When describing the value of an element
When you want to describe a value of an element, describe the value along with the data type of the element. For example, a comment that is to indicate a value of an element handled in decimal format, such as UINT16, is written in decimal format, and a comment that is to indicate a value so that it is handled in hexadecimal format, such as WORD, is written in hexadecimal format (with "0x" at the start of the value). Further, for methods and messages, describe the meaning of each value that is to be read and written, the meaning of the value of the error register, and the like, as needed.

(2) When describing the value of an element in which the item MIN_INC is described
When describing in a comment the value of an element in which the item MIN_INC is described, describe the value based on one of the following:

[1] Describe the value based on the value prior to calculation by the item MIN_INC.

[2] Describe the value based on the unit and setup specified in the item ENG_UNIT.
For example, if you want to write a comment for an element in which the RANGE item is [0,5000], the MIN_INC item is 0.01, and the ENG_UNIT item is %, either write "Set 2500 or higher when function A is used" using the description method of 1 above, or write "Set the value to 25% or higher when function A is used" using the description method of 2 above. Note, however, that the description method of 1 should always be used for an element for which the utility software cannot express the item MIN_INC or ENG_UNIT (in a case where "NA" is written for the item ENG_UNIT or MIN_INC of an element of a COMM_IF section serving as a reference source).

(3) When describing a parameter, method, or message that is potentially not applicable, depending on the device version
For a parameter, method, or message that is potentially not applicable with older devices, we recommend that you describe the applicable (or non-applicable) version information in the COMMENT item.

(4) When describing multiple types of comments for one element
Basically, describe a comment for one element with one item. However, in a part such as ENUM (see Section 5.3.9), which has the specification in which comments with multiple meanings can be described for one element, multiple comments can be described. When multiple comments can be described, the meaning of each comment is clearly specified in the specification.

[Note] The comments can be described only in CC-Link IE TSN.

4.3.1.7 Conventions of CONDITIONx

Write in CONDITION part, and write the range of conditions in order to conditionally determine the value of elements referenced from item CONDITIONXVARIABLE. For detailed conventions, see Section 5.4.9.

(1) Use with RESULTx (See Section 4.3.1.40) in the conditionMember division of CONDITION part (See Section 5.4.9).

(2) A maximum of 32 CONDITIONx items can be written in 1 conditionMember division.

4.3.1.8 Conventions of CONDITIONXVARIABLE

Write in CONDITION parts, and write references to elements used in conditional determination. For detailed conventions, see Section 5.4.9.

(1) Use in the conditionVariable division of CONDITION part (See Section 5.4.9).

(2) A maximum of 32 CONDITIONXVARIABLE items can be written in 1 conditionVariable division.

4.3.1.9 DATA conventions

A DATA item describes the contents of the element. Describable values are in accordance with the specifications of the element defined in the parts FILE_INFO, DEVICE_INFO, COMM_IF_INFO, and BLOCK_INFO. If not clearly defined, the description range is equivalent to that for the DATATYPE item of the element. For details, see Sections 5.1.1.3, 5.2.1.3, 5.3.1.3, and 5.4.1.3.
When other languages are supported, write this label name in keeping with the descriptions of the LANGUAGE element and FILE_INFO part of the FILE section. For CC-Link IE TSN, the specific element of COMM_IF_INFO part can be described in array type. Place the value in brackets (see Section 4.3.3) and describe multiple contents of the element.
[Note] The description in which the value described between "<" and ">" can be used only in CC-Link IE TSN.

4.3.1.10 DATATYPE conventions

A DATATYPE item describes the data type of the value corresponding to the element. The data type not only indicates the type of the element, but also specifies the memory size occupied when the element is assigned to a module as well. lists the data types that can be specified.
Excluding cases where reference is made to other elements using the REF item (see Section 4.3.1.30), DATATYPE may be omitted.

Table 4-10 Data Type List

Data Type	Expression	Description	Value Range
Bool	BOOL	A bit that indicates the status of two alternatives, such as ON and OFF	0 or 1 0: Indicates FALSE. 1: Indicates TRUE.(*4)
Bit string (Hexadecimal expression)*2	BYTE	A bit array of 8 bits	0x00 to 0xFF
	WORD	A bit array of 16 bits	0x0000 to 0xFFFF
	DWORD	A bit array of 32 bits	0x00000000 to 0xFFFFFFFF
	BIT_STRINGx (x=2 to 15)	A bit array of x bits, where x is an integer between 2 and 15	0x0 to 0x7FFF *(The maximum value differs according to the number of bits of x.)
Signed integer (INT) (Decimal expression)*2	INT8	A signed integer of 8 bits	-128 to 127
	INT16	A signed integer of 16 bits	-32768 to 32767
	INT32	A signed integer of 32 bits	-2147483648 to 2147483647
	INTx (x=2 to 15)	A signed integer of x bits, where x is an integer between 2 and 15	-2^{x-1} to $(2^{x-1}-1)$
Unsigned integer (UINT) (Decimal expression)*2	UINT8	An unsigned integer of 8 bits	0 to 255
	UINT16	An unsigned integer of 16 bits	0 to 65535
	UINT32	An unsigned integer of 32 bits	0 to 4294967295
	UINTx (x=2 to 15)	An unsigned integer of x bits, where x is an integer between 2 and 15	0 to $(2^{x}-1)$
BCD integer (BCD) (Decimal expression)*2	BCD8	An 8-bit BCD type integer	0 to 99
	BCD16	A 16-bit BCD type integer	0 to 9999
	BCD32	A 32-bit BCD type integer	0 to 99999999
	BCDx (x=4,12)	An x-bit BCD type integer, where x is either 4 or 12	Positive integer of (x/4) digits
Real number (Decimal expression) *2	REAL	A 32-bit real number	-3.402823E+38 to -1.175494E-38,0,1.175494E-38 to 3.402823+38 (Number of significant digits: 7)
	LREAL	A 64-bit real number	-1.797693134862315E+308 to -2.225073858507202E-308,0, 2.225073858507202E-308, to 1.797693134862315E+308 (Number of significant digits: 15)
String*3	STRING(x) (x = An integer greater than or equal to 1)	Indicates a string (alphanumeric characters or symbols) that can be expressed in ASCII code	The maximum value differs according to the value of x.
	STRING_U(x) (x = An integer greater than or equal to 1)	Indicates a string based on Unicode (utf-8)	The maximum value differs according to the value of x.
Set*3	Data type + "()" (Example: INT8(),STRING(10)())	Indicates a set type. Using a set type enables enumerated descriptions of multiple sets of information for the item DATA (see Section 4.3.1.9).	The range of the value of each set element is in accordance with the data type.
Structure type	STRUCT	Structure type	*1
Constant Type	"CONST"+" " +Data Type [Ex] CONST INT8, CONST WORD"	Shows a constant value. Used when it is necessary to read/write constants in METHODS and MESSAGES. Cannot be used in FILE_INFO, DEVICE_INFO, BLOCK_INFO, COMM_IF_INFO (Even if CONST is not written, the value of DATA shows constants). Types that can be written in data types in each part are the same as types that can be used alone without writing CONST. However, structural type cannot be written.	Follows the data type written after "CONST".

*1: For a structure type, describe the details of the data structure using the part STRUCT (see Section 5.3.8). This type cannot be used with all elements. Rather, it is only applicable to the elements of the parts COMM_IF_INPUT, COMM_IF_OUTPUT, COMM_IF_PARAMETER, and COMMAND_ARGUMENT (it cannot be written with the parts of a BLOCK section other than the parts STRUCT and COMMAND_ARGUMENT). Further, the data size of a structure type is handled in word units. Even if an offset for the number of bits is used up to F, the data size is raised and handled in word units (example: the size is regarded as 3 even if the offset is used only up to 2.4). Note, however, that, in a case where the size has been clearly defined as handled in bit units (see Section 4.3.1.42), an exception is made and the number of bits used is regarded as the size (example: the number of bits is 36 when the offset is used up to 2.4).
The items DEFAULT, RANGE, MIN_INC, ENG_UNIT, and REF cannot be written in an element of a structure type (in an element where the DATATYPE is STRUCT).

*2: The value of each item (such as DEFAULT, RANGE, COMMENT, etc.) of an element in which the item DATATYPE is an integer type (INT, UINT, BCD) or real number type, is described using a decimal expression. The value of each item of an element in which the item DATATYPE is a bit string type, such as WORD, must be written using a hexadecimal expression (written with "0x" at the start of the value). Note that, when you want to write a hexadecimal value, we recommend that you do not omit high-order zeros in keeping with the data size. (If the type is WORD, we recommend that you do not write 0x3F, but rather 0x003F.)

*3: Not applicable to any parts other than FILE_INFO, DEVICE_INFO, BLOCK_INFO, and COMM_INFO.

*4: A description such as "FALSE," "TRUE," "OFF," or "ON" is not permitted, unless a description method based on "ON" and "OFF" is clearly indicated in the specification, such as for the item INTERLOCK (see Section 4.3.1.18).

*5: Not applicable to any parts other than FILE_INFO, DEVICE_INFO, BLOCK_INFO, COMM_IF_INFO, COMM_IF_PARAMETER, and BLOCK_PARAMETER.

[Note] The DATATYPE item is restricted in terms of specifiable content, according to the element as follows:

(1) With an element of the part COMM_IF_INPUT: When the ASSIGN item has the prefix "RX," only types of the following sizes can be written: 1, 4, 8, 12, 16, and 32 bits.

(2) With an element of the part COMM_IF_INPUT: When the ASSIGN item has the prefix "RWr," only a type of a 16-bit or 32-bit size or a structure type can be written. Note, however, that in a case where a STRUCT part is used for reference purposes, an element distributed across multiple RWrs cannot be written.

(3) With an element of the part COMM_IF_OUTPUT: : When the ASSIGN item has the prefix "RY," only types of the following sizes can be written: 1, 4, 8, 12, 16, and 32 bits.

(4) With an element of the part COMM_IF_OUTPUT: When the ASSIGN item has the prefix "RWw," only a type of a 16-bit or 32-bit size or a structure type can be written. Note, however, that in a case where a STRUCT part is used for reference purposes, an element distributed across multiple RWrs cannot be written.

4.3.1.11 DEFAULT conventions

A DEFAULT item describes the default value of the element. Default values are restricted as follows.

(1) Relationship with DATATYPE item
The default value must be within the range of the data type described in the DATATYPE item (see Section 4.3.1.10). A value outside of the range of the RANGE item (see Section 4.3.1.26) can be written as long as the value is in accordance with the DATATYPE item. For example, for an element with a RANGE item described as 1 to 100, you can write 0, an invalid number, as the default value.
Further, a default value cannot be written when the DATATYPE item is an element of the part STRUCT. In such a case, write the default value in the DEFAULT item of each element defined in the STRUCT part.

(2) Description when default value is undefined
When the default value is not uniquely defined, the description may be omitted.

(3) Description when the default value cannot be indicated based on a specific communication method.
When the default value is undefined based on communication type or communication settings, write "NA" in the element of the COMM_IF section.

4.3.1.12 END_CONDITION conventions

An END_CONDITION item describes the condition for ending reading (writing) based on a conditional expression. Note that when a process is to end upon satisfaction of the conditions of the item REQ_FLAG (see Section 4.3.1.34), this item may be omitted. The basic conventions for this item are the same as those for the INTERLOCK item (see Section 4.3.1.18), but the item may also include the following description as well:

(1) When wait time is required prior to end assessment, write WAIT(X) prior to the conditional expression and separated by a comma. X is any integer between 1 and 65535, and indicates that a wait time of X ms is required. A comma cannot be entered at any location other than immediately after WAIT(X).
(Example: WAIT(1000),RX8==ON),
(Example: WAIT(1000)),
(Example of unpermitted description: RX8==ON,WAIT(1000))
* The utility software does not need to guarantee that the conditional expression will be assessed the instant that X ms elapse. The conditional expression is assessed after X ms elapse.
* While not described in the ERR_CONDITION item, error assessment is also performed after waiting X ms as well.

4.3.1.13 ENG_UNIT conventions

An ENG_UNIT item describes a string that indicates the unit of measurement. For example, when you want to indicate the duration of a second, write "s". The following describes the description specifications.

(1) This item may be omitted when an engineering unit is not needed.

(2) The item can be written using any string.

(3) When the engineering unit cannot be indicated when a specific communication method is used, enter "NA" in the ENG_UNIT item of the element of the COMM_IF section.

(4) This item cannot be described when the item DATATYPE (see Section 4.3.1.10) is an element of the part STRUCT. Describe the engineering unit in the item ENG_UNIT (see Section 4.3.1.26) of each element defined in the part STRUCT.

(5) Further, a default value cannot be written when the DATATYPE item (see Section 4.3.1.10) is an element of the STRUCT part. In such a case, write the default value in the ENG UNIT item (see Section 4.3.1.26) of each element defined in the STRUCT part.

When other languages are supported, write this label name in keeping with the descriptions of the LANGUAGE element and FILE_INFO part of the FILE section.

4.3.1.14 ERR_CONDITION conventions

An ERR_CONDITION item describes an abnormal end flag during read (write) command execution using a conditional expression. Note that, for processing in which an error does not occur [in a case

where the conditions of the END_CONDITION item (see Section 4.3.1.12) are always satisfied], this item may be omitted. The conventions are the same as those for the INTERLOCK item (see Section 4.3.1.18).

4.3.1.15 ERR_CODE RANGE conventions

An ERR_CODE RANGE item describes the range of an error code. While the conventions are the same as those for the RANGE item (see Section 4.3.1.26), indicate correspondence with the string that indicates the meaning of the value of the error code basically by using the ENUM part description. Note that omission of the description of the ERR_CODE_RANGE item indicates that all values that can be expressed using endCode within the error PDU specified by the ERR_TYPE item (for the item of a MESSAGE part; see Section 4.3.1.17) or any device specified by the ERR REGISTER item (for the item of a METHOD part; see Section 4.3.1.16) are within the error code range.

4.3.1.16 ERR_REGISTER conventions

An ERR_REGISTER item describes the device name that stores error codes when an error occurs. Write the device name following the conventions below.

(1) Write the name using the same conventions as those for the ASSIGN item (see Section 4.3.1.3).

(2) Write the name by referring to the ASSIGN item of the element using a reference specification (see Section 4.3.3) by "$()".

If not needed, the ERR_REGISTER item may be omitted.

4.3.1.17 ERR_TYPE conventions

An ERR_TYPE item describes the type of the data format to be used in response processing when an error occurs.
Write a string that connects the error PDU name and ASCII or Binary using "_". For the error PDU name, see "PDU name" in Table 4-19. [The table is an extraction from the "CC-Link IE Field Network Specification (SLMP Specification Overview) BAP-C1605-005-A." For details of the contents described in the table, see the "PDU List" in the "CC-Link IE Field Network Specification (SLMP Specification Overview) BAP-C1605-005-A."] ASCII and Binary respectively indicate that the data is to be sent in ASCII mode or Binary mode. This item may be omitted when errors do not occur (when error information is not required).

[Description example] rdErrST_Binary,wrErrMT_ASCII

Table 4-11 Error PDU Name List

Type	SLMPPDU	PDU Name
ST	rdErrST-PDU	rdErrST
	wrErrST-PDU	wrErrST
MT	rdErrMT-PDU	rdErrMT
	wrErrMT-PDU	wrErrMT

4.3.1.18 INTERLOCK conventions

INTERLOCK describes an interlock using a conditional expression. The description specifications are described below. This item may be omitted if an interlock does not exist (if there are no start conditions).

(1) For the term on the left of the operator, directly write the address of the register name, for example, or use a reference specification via "$()" (see Section 4.3.3) to refer to the ASSIGN item (see Section 4.3.1.3) of the element.

(2) If you describe the interlock by a reference using "$()" in the left term, the data type of the left term is in accordance with the description of the DATATYPE item of the element of the reference. If you directly write a word device, the data size is regarded as 16 bits, and the type category is in accordance with the value of the right term.

(3) Make sure that the data sizes and type categories of the left and right terms coincide.

(4) The relational operators shown in Table 4-12 can be used operators.

Table 4-12 Applicable Relational Operators

Operator	Description
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

(5) If you want to execute a bit device comparison, write "ON" or "OFF" in the right term.
(Example: RYC==ON)

(6) If multiple conditions need to be satisfied, write the conditions side-by-side using the conjunction operator "&&".
(Example: RYC==ON&&RXE==OFF)

4.3.1.19 LABEL conventions

A LABEL item is used to refer to the element. Write a label name that identifies the section, part, or element. The LABEL item can be written using a 32-character Unicode alphanumeric string. Table 4-13 shows the characters that are cannot be used.
The following restrictions apply to the label name written using the LABEL item:

(1) The label name of a section needs to be written so that there are no duplicates.

(2) The label name of a part needs to be written so that there are no duplicates within the same section.

(3) The label name of an element needs to be written so that there are no duplicates within the same part.

(4) Label name comparisons are assessed without upper/lower case sensitivity ("VAR1" and "var1" are regarded as duplicate names).

(5) If multiple COMM_IF INPUT or COMM_IF_OUTPUT parts containing cyclic address information exist in the same section, the label names need to be written so that there are no duplicates among the elements of the different parts.

The LABEL item is not only used when you want to refer to other elements within a profile description, but also when the utility software deploys a sequence program label name based on the profile description.
When elements are compared across profile descriptions having the same model name but different CSP+ specification versions, file versions, device versions, or supported language information, the elements are regarded as having the same meaning if the LABEL items of the sections, the LABEL items of the parts, and the LABEL items of the elements are the same. For this reason, caution is required when profile descriptions are to be changed as a result of version upgrades.

Table 4-13 Characters Not Permitted in LABEL and LABEL2 Items

Category	Prohibited Characters
Characters not permitted in LABEL item	!, ", #, $, %, &, ', (,), *, +, , , -, ., /, :, ;, <, =, >, ?, @, [, ¥,], ^, `, {, |, }, ~
Characters not permitted at head of LABEL item	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, _
Reserved words	• ACCESS, AUTO_PARAMETER, ARGUMENT, ASSIGN, BLOCK, BLOCK_COMMAND, BLOCK_INFO, BLOCK_INPUT, BLOCK_OUTPUT, BLOCK_PARAMETER, CATEGORY, CODE, COMM_IF, COMM_IF_COMMAND, COMM_IF_INFO, COMM_IF_INPUT, COMM_IF_OUTPUT, COMM_IF_PARAMETER, COMMAND, COMMAND_ARGUMENT, COMMENT, COMMENTx (x = An integer greater than or equal to 1), COMMON, CONDITION, CONDITIONx (x = An integer greater than or equal to 1), DATA, DATATYPE, DEFAULT, DEVICE, DEVICE_INFO, DUMMY, END_CONDITION, ENG_UNIT, ENUM, ERR_CODE_RANGE, ERR_CONDITION, ERR_REGISTER, ERR_TYPE, FILE, FILE_INFO, INTERLOCK, LABEL, LABEL2, MAINTENANCE, MESSAGE, MESSAGE_TYPE, METHOD, METHOD_TYPE, MIN_INC, MONITOR, NA, NAME, OFFSET, OTHER, PARAMETER, PARENT_TARGET, RANGE, READ_DATA, READ_DATATYPE, READ_REGISTER, REF, RELATED_ELE, RELATED_MESSAGE, RELATED_METHOD, REQ_FLAG, REQUEST_DATA, REQUEST_DATATYPE, REQUEST_TYPE, RESPONSE_DATA, RESPONSE_DATATYPE, RESPONSE_TYPE, RESULTx (x = An integer greater than or equal to 1), SEQ_TARGET, STRUCT, TARGET, UI_ATTRIBUTE, VALUE, WRITE_DATA, WRITE_DATATYPE, WRITE_ORDER, WRITE_REGISTER • The above strings cannot be used, regardless of the upper/lower case setting. (Example: LABEL and laBel and label are all reserved words.) • Any string that fully matches one of the above is not permitted. (Example: A_LABEL and LABELA can be used.) • **Strings that start with "P_" are reserved for future expansion.** • **Strings that start with "SLMP_P_" are reserved for SLMP message use.**
Space	Both single-byte and double-byte spaces are not permitted.

Note that the applications of the regulated label names described in Section 5.5 are already determined. The names cannot be used for any other application.
Further, label names that start with "P" will be used in the future for extending descriptions common to modules defined in Section 5.5. Do not, therefore, use the label names that start with "P" with labels of elements or parts that describe module specific functions.

4.3.1.20 LABEL2 conventions

A LABEL2 item describes the label name used for element identification in keeping with supported language information. The LABEL2 item can be written using a 32-character Unicode string (including characters other than alphanumeric as well). Table 4-13 shows the characters that are cannot be used. If the LABEL item and the LABEL2 item have the same descriptions, the LABEL2 item may be omitted. The following restrictions apply to the label name written using the LABEL2 item:

(1) The label name of an element needs to be written so that there are no duplicates within the same part.

(2) Label name comparisons are assessed without upper/lower case sensitivity ("VAR1" and "var1" are regarded as duplicate names).

(3) If multiple COMM_IF INPUT or COMM_IF_OUTPUT parts containing cyclic address information exist in the same section, the label names need to be written so that there are no duplicates among the elements of the different parts.

The LABEL2 item is sometimes used when the utility software deploys a name as a sequence program label name based on the profile description.

When you want to describe the same elements when profile descriptions are changed due to a version upgrade, we recommend that you do not change the contents of the LABEL item as well as the LABEL2 item. Note, however, that sometimes elements are assessed as the same if simply the LABEL items

match, depending on the utility software.
When other languages are supported, write this label name in keeping with the descriptions of the LANGUAGE element and FILE_INFO part of the FILE section.

4.3.1.21 MESSAGE_TYPE conventions

A MESSAGE_TYPE item describes the message type by describing a TYPE value of Table 4-14. The MESSAGE_TYPE item cannot be omitted. If you want to write multiple TYPE values, write them as a set (see Section 4.3.3, Table 4-23 [2]).
This description is used to assess whether the data is to be displayed in the parameter settings of general-purpose utility software, displayed at command execution, or not displayed.

Table 4-14 Values of MESSAGE_TYPE Item

No.	TYPE Value	Meaning
1.	PARAMETER	Indicates that the message is a parameter related message.
2.	COMMAND	Indicates that the message is a command related message.
3.	MONITOR	Indicates that the message is processed not just once but consecutively. As for the method, the MONITOR type does not mean that the method processing is executed consecutively all the time. As for the message, however, the message is always executed consecutively. Note that only the corresponding message is executed consecutively. (If "PRE" and "POST" are described in RELATED_MESSAGE, those messages are executed once before and after the consecutive execution of the corresponding message.)
4.	MAINTENANCE	Indicates that the message is a maintenance related message. If the message is to acquire maintenance information, information may be acquired periodically. Note the following. ・ When "MONITOR" is defined to MESSAGE_TYPE as a set, information is acquired according to the operating specifications of the MONITOR type.
5.	OTHER	Indicates a message that is not displayed by general-purpose utility software. Also describes a method that has significance for the first time after another method is called by a RELATED_MESSAGE item.
6.	AUTO_PARAMETER	Indicates that the message is a message for slave station parameter auto setting in CC-Link IE TSN. **[Note]** Available only in CC-Link IE TSN.

4.3.1.22 METHOD_TYPE conventions

A METHOD_TYPE item describes the type of the method by describing a TYPE value of Table 4-15. The METHOD_TYPE item cannot be omitted. If you want to describe multiple TYPE values, describe them as a set (see Section 4.3.3, Table 4-23 [2]).
This description is used to assess whether the data is to be displayed in the parameter settings of general-purpose utility software, displayed at command execution, or not displayed.

Table 4-15 Values of METHOD_TYPE Item

No.	TYPE Value	Meaning
1.	PARAMETER	Indicates that the method is a parameter related method.
2.	COMMAND	Indicates that the method is a command related method.
3.	MONITOR	Indicates that the method is processed not just once but consecutively. This does not mean that the method processing is executed consecutively all the time. This means that processing that can obtain effect is executed consecutively. For example, as for the method that reads values in the result output register, the method itself is not executed consecutively, but the read processing is executed consecutively. Note that only the corresponding method is executed consecutively. (If "PRE" and "POST" are described in RELATED_METHOD, those methods are executed once before and after the consecutive execution of the corresponding method.)
4.	MAINTENANCE	Indicates that the method is a maintenance related method. If the method is to acquire maintenance information, information may be acquired periodically. Note the following. • When "MONITOR" is defined to METHOD_TYPE as a set, information is acquired according to the operating specifications of the MONITOR type.
5.	OTHER	Indicates a method that is not displayed by general-purpose utility software, such as a method that has significance, for example, when utilized with parameter setup or a ladder in the case of the AD converter initial setup function, but not when displayed by general-purpose utility software. Also describes a method that has significance for the first time after another method is called by a RELATED_METHOD item.

4.3.1.23 MIN_INC conventions

A MIN_INC item describes the minimum resolution value separately from the ENG_UNIT item (see Section 4.3.1.13). The description specifications are as follows:

(1) This item cannot be described for elements in which an ENG_UNIT item is not described.

(2) You can write the item for an element in which the DATATYPE item (see Section 4.3.1.10) is a signed integer type (INT), an unsigned integer type (UINT), or a BCD integer type (BCD) (elements that treat values in decimal format).

(3) The writable value is 10^n (where n is an integer); write the calculation result in decimal format (. . ., 0.01, 0.1, 1, 10, 100, . . .). Note that an element with the MIN_INC item omitted is treated as if the minimum increment is 1.

(4) In a case where the MIN_INC item cannot be indicated when a specific communication method is used, write "NA" in the MIN_INC item of the element of the COMM_IF section.

(5) When using a specific communication system, if MIN_INC items cannot be shown, write "NA" in the MIN_IC item of elements of the COMM_IF section.

The values of elements of the RANGE item (see Section 4.3.1.26), DEFAULT item (see Section 4.3.1.11), and the like are values maintained by the module. The value of the ENG_UNIT item can be acquired by multiplying such values by the MIN_INC item.

For example, in a case of the element described below, the value maintained by existing equipment is in the range of 1000 to 2000. However, since 0.01 is written in the MIN_INC item, the actual setting range becomes 10.00 Hz (0.01 Hz x 1000) to 20.00 Hz (0.01 Hz x 2000). The following shows an example of an element in which the minimum increment is described.

```
<blockInputMember label="SetingFreq">
   <label2><item> Output frequency </item></label2>
   <name><item> Output frequency </item></name>
   <datatype><item>UINT16</item></datatype>
   <default><item>0</item></default>
   <range><item>[1000,2000]</item></range>
   <minInc><item>0.01</item></minInc>
   <emgUnit><item>Hz</item></engUnit>
</blockInputMember>
```

4.3.1.24 NAME conventions

A NAME item describes the name of the element. The written string is used to display the name or contents of the element by utility software.
The NAME item cannot be omitted, unless the element is an element that refers to another element using the REF item described later (see Section 4.3.1.30) or an element of the FILE_INFO, DEVICE_INFO, COMM_IF_INFO, or BLOCK_INFO part.
When other languages are supported, write this label name in keeping with the descriptions of the LANGUAGE element and FILE_INFO part of the FILE section.

4.3.1.25 OFFSET conventions

An OFFSET item describes the offset assigned within the structure. The following describes the rules for offset descriptions:

(1) Write the description of the OFFSET item as "NumberOfWords.NumberOfBits" with the offset for the number of words (hexadecimal number) and the offset for the number of bits (hexadecimal number) separated by a dot. (Example: 0.1, 2.D, F.3)

(2) The sequence of the elements within a part does not have to match the order of the offset values described for the elements.

(3) The OFFSET item cannot be omitted.

(4) Duplication of an offset in elements within the same STRUCT part is prohibited. (Example: Given a data type of UINT4 and elements with the offsets "0.0" and "0.3," bit 3 is duplicated, and thus the description is not permitted.)

(5) An empty offset is permitted. (Example: A STRUCT description having the offsets "0.0," "0.2," and "0.8" with all data types as BOOL is permitted.)

4.3.1.26 RANGE conventions

A RANGE item describes the range of a value to be set for the element. The following describes the description method of the setting range.

(1) When directly describing a value and value range
For a value range that describes a "value" and the "value range" one after the other, write the two values of the value range within brackets and separate the two values with a comma (","). The left value indicates the low limit, and the right value indicates the high limit. A closed interval is written using "[","]" and an open interval is written using "(",")". If there are multiple value ranges, list the ranges one after the other as a set.
Example: "[1,600]" "[7000,7010]" "8800"
With an XML file: (<item>[1,600]</item><item>[7000,7010]</item><item>8800</item>)

(2) When using an ENUM part
With an ENUM part, you can indicate the setting range by describing the available options in the ENUM part and writing a reference to the ENUM part (see Section 5.3.9).
Use an ENUM part if you want to determine the value selected via a list box or pull-down menu of the utility software, or if you want to display the meaning of each value during monitoring.
Note, however, that an ENUM part can only be used if the DATATYPE item (see Section 4.3.1.10) is a Boolean type , integer type (INT, UINT, BCD), or a bit string type (BYTE, WORD, etc.).

(3) When the range is equal to the range of the value for the DATATYPE item
When the setting range is equal to the range of the DATATYPE item, the RANGE item may be omitted.

(4) When the setting range cannot be displayed (does not need to be displayed) by utility software
When the RANGE item is handled by utility software and does not need to be displayed (cannot be displayed), write "NA".

(5) When writable values differ from readable values
Read and write attributes cannot be set for each value written in a RANGE item.
As a result, an element for which the readable values are 0 to 10 and the writable values are 0 to 5, for example, cannot be expressed using the RANGE item.
In such a case, we recommend writing [0,10] for the RANGE item, and writing that 6 and above are not writable in the COMMENT item (see Section 4.3.1.6).

(6) When the DATATYPE item is a structure type
In this case, a RANGE item cannot be written. Write the values in the RANGE item of each element defined in the STRUCT part.

(7) When a DATATYPE item is structural type
RANGE items are not permitted. Write in the RANGE items of various elements defined in STRUCT part.

(8) When the set values of other parameters are used as the upper limit or lower limit of the setting range
Using the conventions in Table 4-23 (3) (reference destination specification by "$()"), reference the VALUE of other parameters to write. Not only maximum value and minimum value, but also descriptions such as "Over 1 but less than parameter A, or over parameter B but less than 1000" are permitted.
This convention can only be written when CC-Link IE Safety Communication Function Support Unit.
This convention is only usable with elements of BLOCK_PARAMETER part, and reference is limited to elements of those parts. (Parameter values can only show the setting range of parameters) Cannot reference anything except VALUE, such as DATA).
Ex 1: When the setting range is greater than 1, but less than the set value of ParamA, write as [1,$(ParamA.VALUE)]
Ex 2: When the setting range is greater than 1 but less than ParamA, or greater than ParamB but less than 1000, write as "[1,$(ParamA.VALUE)),[$(ParamB.VALUE),1000)"

4.3.1.27 READ_DATA conventions

A READ_DATA item indicates the value to be read, and is described following the conventions below. Note that the value specified needs to comply with the data type of the READ_DATATYPE item (see Section 4.3.1.28).

(1) Specify the value by writing a reference to the VALUE held by the element of the TARGET item (see Section 4.3.1.42), the size of the DATATYPE item (see Section 4.3.1.10), the ASSIGN item (see Section 4.3.1.3), and the element of the COMMAND_ARGUMENT part, placing the reference between "$(" and ")" (see Section 4.3.3).
(Example: $(VALUE),$(ARGUMENT.ReadFrequencyData.VALUE)
* When specifying an element of COMMAND_ARGUMENT, you cannot specify an element that is not readable (NA, W, etc.) by the ACCESS item (see Section 4.3.1.1).
* When describing the reading of all elements inside TARGET using an asterisk (example: $(*.VALUE)), you cannot describe an element that is not readable in TARGET.
* When multiple elements are referenced in TARGET and one of the elements is to be selected and read, you can write an element that is not readable in TARGET.
* In an element in which the METHOD_TYPE is PARAMETER and VALUE is referenced by WRITE_DATA, you cannot describe a VALUE reference in READ_DATA.

(2) If multiple read registers have been written using the READ_REGISTER item (see Section 4.3.1.47), specify the same number of write values as read registers, placing the READ_DATA item between "<" and ">" (see Section 4.3.3) as well.
(Example: <0x40><$(VALUE)>,<$(*.VALUE)>)

(3) This item cannot be omitted if the READ_REGISTER item is omitted.

4.3.1.28 READ_DATATYPE conventions

A READ_DATATYPE item indicates the data type of the data type to be read, and is written following the conventions below.

(1) Write the data type directly (see Section 4.3.1.10).

(2) Specify the data type by writing a reference to the DATATYPE item (see Section 4.3.1.10) of the element, placing the reference between "$(" and ")" (see Section 4.3.3).
(Example: $(DATATYPE))
* A reference cannot be made via a LABEL item (see Section 4.3.1.19) of an element in which a TARGET item (see Section 4.3.1.42) is not described.
* Specification of an element or argument in which the ACCESS item (see Section 4.3.1.1) is not readable ("NA", "W", etc.) is not permitted. Note, however, that writing a description that refers to multiple elements using the TARGET item and including an element that is not readable is unproblematic.

(3) When you want to write multiple read registers using the READ_REGISTER item (see Section 4.3.1.29), specify the same number of data types as read registers, placing the READ_DATATYPE item between "<" and ">" (see Section 4.3.3) as well.
(Example: <WORD><$(DATATYPE)>,<$(*.DATATYPE)>)

(4) This item cannot be written if the READ_REGISTER item is omitted.

4.3.1.29 READ_REGISTER conventions

A READ_REGISTER item specifies the register to be read using the conventions below. The read value needs to comply with the READ_DATA item (see Section 4.3.1.27).

(1) Directly describe the register using a string. [For the conventions, see the ASSIGN item (see Section 4.3.1.3).]
(Example: RWr1)

(2) Specify the register by referring to the ASSIGN item of the element via a reference specification using "$(ASSIGN)" in a case where a reference has been described in the TARGET item (see Section 4.3.1.42). (No item other than ASSIGN can be used as reference.)
(Example: $(InitParameters.CH1_AvgTime.ASSIGN), $(ARGUMENT.ReadFrequencyData.ASSIGN))

(3) If you want to specify multiple registers to be read, write the register names in order, placing each name between "<" and ">" (see Section 4.3.3).
(Example: <RWr4><$(ASSIGN)>, <$(*.ASSIGN)>)
Note that, if multiple registers are described, reading is performed in order, starting from the register written on the left.

(4) In a case where there are multiple content descriptions in the corresponding READ_DATA item, including content that does not need to be read, write the content description of the READ_REGISTER item corresponding to the content that does not need to be read as $(DUMMY). The content of the READ_DATA item corresponding to $(DUMMY) will not be read from the register. (Example: In a case where three content descriptions exist in the READ_DATA item and the first content does not need to be read, write the READ_REGISTER item as follows: <$(DUMMY)><RWr2><RWR3>.)

(5) This item may be omitted if the method does not require reading.

4.3.1.30 REF conventions

A REF item describes a reference from the COMM_IF section to the BLOCK section. For reference conventions, see Section 4.3.4.

(1) For a COMM_IF_INPUT part, write a reference to the corresponding BLOCK_OUTPUT element.

(2) For a COMM_IF_OUTPUT part, write a reference to the corresponding BLOCK_INPUT element.

(3) For a COMM_IF_PARAMETER part, write a reference to the corresponding BLOCK_PARAMETER element.

(4) For a COMM_IF_COMMAND part, write a reference to the corresponding BLOCK_COMMAND element. When an ARGUMENT item (see Section 4.3.1.2) is written, the REF item cannot be described.

(5) For a STRUCT part, write a reference to be referred to by the element. Such a reference is used to refer to the inputs/outputs and parameters of the BLOCK section from each element of the structure. The referable elements differ according to the reference source of the STRUCT part.
- For a STRUCT part in which a reference is made via the DATATYPE item (see Section 4.3.1.10) of the COMM_IF_INPUT part, it is possible to describe a reference to an element of the BLOCK_OUTPUT part using the REF item.
- For a STRUCT part in which a reference is made via the DATATYPE item of the COMM_IF_OUTPUT part, it is possible to describe a reference to an element of the BLOCK_INPUT part using the REF item.
- For a STRUCT part in which a reference is made via the DATATYPE item of the COMM_IF_PARAMETER part, it is possible to describe a reference to an element of the BLOCK_PARAMETER part using the REF item.
- For a STRUCT part in which a reference is made via the DATATYPE item of COMMAND_ARGUMENT, this item cannot be written.

(6) If a reference to a STRUCT part is described in a DATATYPE item of an element that has a REF item, a reference cannot be written in the REF item.

(7) If the same items exist in the reference source and reference and the values of the items in the reference source and reference are the same, write the value in the element of the reference, and omit the item in the element of the reference source.

(8) If the same items exist in the reference source and reference and the values of the items in the reference source and reference differ, write the value in both the elements of the reference source and reference. In such a case, the utility software uses the value of the reference source.

(9) If multiple COMM_IF sections are written and the setting range, access attributes, etc., differ according to the COMM_IF section used, you can overwrite the value written in the reference by writing the value in the element of the reference source.

(10) It is possible to describe a reference from a single element to multiple elements (a one-to-many reference) or a reference from multiple elements to one element (a many-to-one reference). A one-to-many reference indicates that, when the inputs, outputs, and parameters indicated by the element of the reference source are accessed, the outputs, inputs, and parameters indicated by all elements of the reference are accessible. A many-to-one reference indicates that the outputs, inputs, and parameters of the reference are accessible from the inputs, outputs, and parameters indicated by all elements of the reference source. The following lists the precautions related to one-to-many and many-to-one references.
- One-to-many reference
  When the item of the element of the reference source is omitted, write the same value for the item of all reference elements.
- Many-to-one reference
  We recommend that you do not use a many-to-one reference in any case other than the following:

[1] A many-to-one reference from different COMM_IF sections

[2] A many-to-one reference from elements of different METHOD or MESSAGE parts

4.3.1.31 RELATED_ELE conventions

A RELATED_ELE item describes a reference from the ENUM part to the COMMAND_ARGUMENT part. If the reference is not required, this can be omitted. For reference conventions, see Section 4.3.4.
[Note] A RELATED_ELE item can be used only in CC-Link IE TSN.

4.3.1.32 RELATED_MESSAGE conventions

A RELATED_MESSAGE item describes a message required as message pre-processing, using the conventions indicated in Table 4-16.

(1) In order to share a TARGET item (see Section 4.3.1.42) in a MESSAGE part of a reference source and reference of a RELATED_MESSAGE item, you need to write a "PARENT_TARGET" in the TARGET item of the element of the reference. If you want to describe selection from a plurality in a TARGET item of the element of the reference source, always select and execute the same element using the MESSAGE part of the reference source and reference.

(2) If you want to specify multiple MESSAGE parts, write the convention shown in Table 4-16 in multiple, placing "<" and ">" (See Section 4.3.3) around each. The maximum number of MESSAGE parts specifiable is sixteen.

(3) A RELATED_MESSAGE item cannot be written with a message of a reference from a RELATED_MESSAGE item.

(4) With a reference message or reference source message from a RELATED_MESSAGE item, reference can be made to a VALUE of an element written in a TARGET item or a VALUE of an element of a COMMAND_ARGUMENT part. However, each message related to the RELATED_MESSAGE item must be in accordance with the following:
   (a) You cannot refer to a VALUE of the same element using both the REQUEST_DATA item (see Section 4.3.1.35) and the RESPONSE_DATA item (see Section 0).
   (b) With the REQUEST_DATA item, you can refer to a VALUE of the same element multiple times.
   (c) With the RESPONSE_DATA item, you cannot refer to a VALUE of the same element multiple times.
   (Example: When "<PRE MessageB><PRE MessageC>" are written in the item RELATED_MESSAGE of MessageA, MessageA is executed after the sequential execution of MessageB and MessageC.)
   (d) VALUE descriptions need to be unified to either the write side (REQUEST_DATA) or the read side (RESPONSE_DATA). Only in a case where the message that describes VALUE involves only one element can the descriptions be made on both the write side and the read side.
   (e) The part types of elements targeted by VALUE (elements described in TARGET) need to be unified. For example, you cannot mix a message that targets the VALUE of an element of COMM_IF_PARAMETER and a message that targets the VALUE of an element of COMMAND_ARGUMENT (COMM_IF_COMMAND) and refer to the data using SEQ.
   (f) When the elements of COMM_IF_INPUT, COMM_IF_OUTPUT, and COM_IF_PARAMETER are described in TARGET of each message, element duplication is prohibited (duplication is only possible with COMM_IF_COMMAND). Duplication is prohibited not only for elements of COMM_IF, but also for elements of BLOCK that arereferenced using the item REF from an element of COMM_IF.

(5) If you want to indicate a message to be executed as message pre-processing, describe the message using the keyword "PRE". To share TARGET between the reference source and reference destination messages of RELATED_MESSAGE, the "PARENT_TARGET" needs to be described in the TARGET of the element of the reference destination (see Section 4.3.1.42). If multiple elements are described in TARGET of the element of the reference source, the same elements must always be selected and executed in the reference source and reference destination messages.

(6) If you want to indicate a message to be executed as message post-processing, describe the message using the keyword "POST". To share TARGET between the reference source and reference destination messages of RELATED_MESSAGE, the "PARENT_TARGET" needs to be described in the TARGET of the element of the reference destination (see Section 4.3.1.42). If multiple elements are described in TARGET of the element of the reference source, the same elements must always be selected and executed in the reference source and reference destination messages.

(7) If you want to indicate the consecutive processing of a message, describe the processing using the keyword "SEQ". The description method in a case where the keyword "SEQ" is used is described below.

(a) "SEQ" can be described only with message elements in which "SEQ_TARGET" is described in TARGET. This message is used to describe the definition of the consecutive processing only, and not communication processing.

(b) The MESSAGE_TYPE of the consecutive processing in general is in accordance with the MESSAGE_TYPE of the message that describes "SEQ" in RELATED_MESSAGE. Note that the type does not need to match the MESSAGE_TYPE of the message referenced by SEQ.

(c) A message in which multiple elements are described in TARGET and one element is selected and targeted from the described elements cannot be referenced by "SEQ" (example: a message in which $(VALUE) and $(DATATYPE) are described in each item cannot be referenced). Reference can be made using "SEQ" for descriptions that target all elements of TARGET (example: $(*.VALUE) and messages in which a reference ($(ARGUMENT.Element.VALUE)) to an element of COMMND_ARGUMENT is described.

Table 4-16 RELATED_MESSAGE Item Conventions

Keyword	Meaning	Conventions	Example
PRE	Indicates a message executed as message pre-processing.	""PRE" + " " (single-byte space) + Message label name"	When "<PRE MessageB><PRE MessageC> is described in the RELATED_MESSAGE of MessageA → After MessageB and MessageC are executed in that order, MessageA is executed.
POST	Indicates a message executed as message post-processing.	""POST" + " " (single-byte space) + Message label name"	When "<POST MessageB><POST MessageC> is described in the RELATED_MESSAGE of MessageA → After MessageA is executed, MessageB and MessageC are executed in that order.
SEQ	Indicates that multiple messages are consecutively executed.	""SEQ" + " " (single-byte space) + message label name"	When "<SEQ MessageB><SEQ MessageC> is described in the RELATED_MESSAGE of MessageA → MessageB and MessageC are executed in that order. * MessageA itself cannot describe communication processing and is therefore not executed.

4.3.1.33 RELATED_METHOD conventions

A RELATED_METHOD item describes a method required as method pre-processing, using the conventions indicated in Table 4-17. The RELATED_METHOD item may be omitted if not required.

(1) In order to share a description of a TARGET item (see Section 4.3.1.42) using a method of the reference source and reference of a RELATED_METHOD item, you need to write a "PARENT_TARGET" in the TARGET item of the element of the reference. If you want to describe selection from a plurality in a TARGET item of the element of the reference source, select and execute the same element using the method of the reference source and reference.

(2) If you want to specify multiple methods, write the convention shown in Table 4-17 in multiple, placing "<" and ">" (See Section 4.3.3) around each. The maximum number of methods specifiable is sixteen.

(3) A RELATED_METHOD item cannot be written with a method of a reference from a RELATED_METHOD item.

(4) With a reference method or reference source method from a RELATED_METHOD item, reference can be made to a VALUE of an element written in a TARGET item or a VALUE of an element of a COMMAND_ARGUMENT part. However, each method related to the RELATED_METHOD item must be in accordance with the following:

(a) You cannot refer to a VALUE of the same element with both the WRITE_DATA item (see Section 4.3.1.44) and the READ_DATA item (see Section 4.3.1.27).

(b) With the WRITE_DATA item, you can refer to a VALUE of the same element multiple times.

(c) With the READ_DATA item, you cannot refer to a VALUE of the same element multiple times. (Example: When "<PRE MethodB><PRE MethodC>" are written in the item RELATED_METHOD of MethodA, MethodA is executed after the sequential execution of MethodB and MethodC.)

(d) VALUE descriptions need to be unified to either the write side (WRITE_DATA) or the read side (READ_DATA). Only in a case where the method that describes VALUE involves only one element can the descriptions be made on both the write side and the read side.

(e) The part types of elements targeted by VALUE (elements described in TARGET) need to be unified. For example, you cannot mix a method that targets the VALUE of an element of COMM_IF_PARAMETER and a method that targets the VALUE of an element of COMMAND_ARGUMENT (COMM_IF_COMMAND) and refer to the data using SEQ.

(f) When the elements of COMM_IF_INPUT, COMM_IF_OUTPUT, and COM_IF_PARAMETER are described in TARGET of each method, element duplication is prohibited (duplication is only possible with COMM_IF_COMMAND). Duplication is prohibited not only for elements of COMM_IF, but also for elements of BLOCK that are referenced using the item REF from an element of COMM_IF.

(5) If you want to indicate a method to be executed as the method pre-processing, describe the method using the keyword "PRE". To share a TARGET between the reference source and reference destination methods of RELATED_METHOD, the "PARENT_TARGET" needs to be described in TARGET of the element of the reference destination (see Section 4.3.1.42). If multiple elements are described in TARGET of the element of the reference source, the same elements must always be selected and executed in the reference source and reference destination methods.

(6) If you want to indicate a method to be executed as the method post-processing, describe the method using the keyword "POST". To share a TARGET between the reference source and reference destination methods of RELATED_METHOD, the "PARENT_TARGET" needs to be described in TARGET of the element of the reference destination (see Section 4.3.1.42). If multiple elements are described in TARGET of the element of the reference source, the same elements must always be selected and executed in the reference source and reference destination methods.

(7) If you want to indicate consecutive processing of a method, describe the processing using the keyword "SEQ". The description method in a case where the keyword "SEQ" is used is described below.

(a) "SEQ" can be described only with method elements in which "SEQ_TARGET" is described in TARGET. This method is used to describe the definition of the consecutive processing only, and not communication processing.

(b) The METHOD_TYPE of the consecutive processing in general is in accordance with the METHOD_TYPE of the method that describes "SEQ" in RELATED_METHOD. Note that the type does not need to match the METHOD_TYPE of the method referenced by SEQ.

(c) A method in which multiple elements are described in TARGET and one element is selected and targeted from the described elements cannot be referenced by "SEQ" (example: a method in which $(VALUE) and $(DATATYPE) are described in each item cannot be referenced). Reference can be made using "SEQ" for descriptions that target all elements of TARGET (example: $(*.VALUE)) and methods in which a reference ($(ARGUMENT.Element.VALUE) to an element of COMMND_ARGUMENT is described.

Table 4-17 RELATED_METHOD Item Conventions

Keyword	Meaning	Conventions	Example
PRE	Indicates a method executed as method pre-processing.	""PRE" + " " (single-byte space) + Method label name"	When "<PRE MethodB><PRE MethodC> is described in the RELATED_METHOD of MethodA → After MethodB and MethodC are executed in that order, MethodA is executed.
POST	Indicates a method executed as method post-processing.	""POST" + " " (single-byte space) + Method label name"	When "<POST MethodB><POST MethodC> is described in the RELATED_METHOD of MethodA → After MethodA is executed, MethodB and MethodC are executed in that order.
SEQ	Indicates that multiple messages are consecutively executed.	""SEQ" + " " (single-byte space) + method label name"	When "<SEQ MethodB><SEQ MethodC> is described in the RELATED_METHOD of MethodA → MessageB and MessageC are executed in that order. * MethodA itself cannot describe communication processing and is therefore not executed.

4.3.1.34 REQ_FLAG conventions

A REQ_FLAG item describes a read (write) request flag using an assignment expression.

(1) Express the assignment operator using "=".

(2) The description specifications for the left term are the same as those for the bit type of the INTERLOCK item (see Section 4.3.1.18).

(3) Write "ON" or "OFF" for the right term. (Example: RYC=ON)

(4) You cannot write information that indicates cancellation of the request flag separately. When the request flag is ON, cancellation is OFF. When the request flag is OFF, cancellation is ON.

(5) If multiple flags need to be set up simultaneously, write the flags separately as a set.
(Example: "RY8=ON" "RY9=ON"
With an XML file: "<item>RY8=ON</item><item>RY9=ON</item>")

When it is acceptable to start processing immediately after the writing of the WRITE_REGISTER item (refer to Section 4.3.1.47) and not cancel the request flag after execution ends, this item may be omitted.

4.3.1.35 REQUEST_DATA conventions

A REQUEST_DATA item describes a value that is to be sent using the data format specified by the REQUEST_TYPE item (see Section 4.3.1.37). The item describes the value that is to be set by a message. The value is written following the conventions described below. Note that the value specified needs to comply with the data type of the REQUEST_DATATYPE item (see Section 4.3.1.36).

(1) Write the value directly. (Example: 0x40)

(2) Specify the value by writing a reference to the VALUE of an element of a TARGET item (see Section 4.3.1.38), the size of a DATATYPE item (see Section 4.3.1.8), an ASSIGN item (see Section 4.3.1.3), and an element of a COMMAND_ARGUMENT part, placing the reference between "$(" and ")" (see Section 4.3.3).
(Example: $(VALUE), <$(ARGUMENT.WriteFrequencyData.VALUE)>, $(DATATYPE).WORDSIZE)

(3) When you want to write multiple values (when you have written a REQUEST_TYPE item so that multiple values are to be described), write the values in list format, placing each between "<" and ">" (see Section 4.3.3).
(Example: <0x40><$(VALUE)>,<$(*.VALUE)>)

(4) Within the configuration of a PDU specified in the REQUEST_TYPE item, write the order of the data other than the header section (command, subCommand, reqData) and the order of the data types in sequence.

(5) If the REQUEST_TYPE item is omitted, or data other than the header section does not exist, omit this item.

(6) When processing that specifies and writes elements of a STRUCT type, which has empty space in the offset, to a REQUEST_DATA item of a message, zero is written to the empty section of the offset. When a fixed value other than zero needs to be written, write an element in which the DATATYPE is constant inside the STRUCT part.

4.3.1.36 REQUEST_DATATYPE conventions

A REQUEST_DATATYPE item describes the data type of a value (REQUEST_DATA) to be sent by request processing, using the conventions below.

(1) Write the data type directly (see Section 4.3.1.10).

(2) Specify the data type by writing a reference to a DATATYPE item (see Section 4.3.1.10) of an element, placing the reference between "$(" and ")" (see Section 4.3.3).
(Example: $(DATATYPE))

(3) When multiple write registers have been written using the REQUEST_DATA item (see Section 4.3.1.35), specify the same number of data types as REQUEST_DATA items, placing each data type between "<" and ">" (see Section 4.3.3).
(Example: <WORD><$(DATATYPE)>,<$(*.DATATYPE)>)

(4) This item cannot be written when the REQUEST_DATA item is omitted.

(5) Within the configuration of a PDU specified in the REQUEST_TYPE item, write the order of the data other than the header section (command, subCommand, reqData) and the order of the data types in sequence.

(6) If the REQUEST_TYPE item is omitted, or data other than the header section does not exist, omit this item.

4.3.1.37 REQUEST_TYPE conventions

A REQUEST_TYPE item describes the type of data format for performing request processing.

Write a string that connects the request PDU name and ASCII or Binary using "_". For the request PDU name, see "PDU name" in Table 4-18. [The table is an extraction from the "CC-Link IE Field Network Specification (SLMP Specification Overview) BAP-C1605-005-A." For details of the contents described in the table, see the "PDU List" in the "CC-Link IE Field Network Specification (SLMP Specification Overview) BAP-C1605-005-A."] ASCII and Binary indicate that the data is to be sent in ASCII mode or Binary mode, respectively.

Although REQUEST_TYPE item must be written in SLMP MESSAGE, MESSAGE other than SLMP can be written for CC-Link IE TSN. In such a case, MESSAGE should be written according to each specifications. For details, see Section 5.5.10.1 and 5.5.10.2.

[Description example] rdReqST_Binary, wrReqMT_ASCII

Table 4-18 PDU Name List for SLMP Request

Type	SLMPPDU	PDU Name
ST	rdReqST-PDU	rdReqST
	wrReqST-PDU	wrReqST
	odReqST-PDU	odReqST
MT	rdReqMT-PDU	rdReqMT
	wrReqMT-PDU	wrReqMT
	odReqMT-PDU	odReqMT

4.3.1.38 RESPONSE_DATA conventions

A RESPONSE_DATA item specifies variables for storing a value to be returned by response processing in the utility software using the data format specified by the RESPONSE_TYPE item (see Section 4.3.1.40), following the conventions below. Note that the value specified needs to comply with the data type of the RESPONSE_DATATYPE item (see Section 4.3.1.39).

(1) Specify the value by writing a reference to VALUE of an element of the TARGET item (see Section 4.3.1.42), placing the reference between "$(" and ")"(see Section 4.3.3).
(Example: $(VALUE), $(ARGUMENT.ReadFrequencyData.VALUE))

When you want to write multiple values (when the RESPONSE_TYPE item is written so that multiple values are to be described), write the values in list format, placing each between "<" and ">" (see Section 4.3.3).
(Example: <0x40><$(VALUE)>, <$(*.VALUE)>)

(2) Within the configuration of a PDU specified in the RESPONSE_TYPE item, write the order of the data other than the header section (command, subCommand, reqData) and the order of the data types in sequence.

(3) If the RESPONSE_TYPE item is omitted, or data other than the header section does not exist, omit this item.

4.3.1.39 RESPONSE_DATATYPE conventions

A RESPONSE_DATATYPE item describes the data type of a value (RESPONSE_DATA) to be returned by response processing, using the conventions below.

(1) Write the data type directly (see Section 4.3.1.10).

(2) Specify the data type by writing a reference to a DATATYPE item (see Section 4.3.1.10) of an element, placing the reference between "$(" and ")" (see Section 4.3.3).
(Example: $(DATATYPE))

(3) When multiple write registers have been written using the RESPONSE_DATA item (see Section 0), specify the same number of data types as RESPONSE_DATA items, placing the RESPONSE_DATATYPE items between "<" and ">" (see Section 4.3.3) as well.
(Example: <WORD><$(DATATYPE)>, <$(*.DATATYPE)>)

(4) This item cannot be written when the RESPONSE_DATA item is omitted.

(5) Within the configuration of a PDU specified in the RESPONSE_TYPE item, write the order of the data other than the header section (command, subCommand, reqData) and the order of the data types in sequence.

(6) If the RESPONSE_TYPE item is omitted, or data other than the header section does not exist, omit this item.

4.3.1.40 RESPONSE_TYPE conventions

A RESPONSE_TYPE item describes the type of data format used in response processing.

Write a string that connects the response PDU name and ASCII or Binary using "_". For the response PDU name, see "PDU name" in Table 4-19. [The table is an extraction from the "CC-Link IE Field Network Specification (SLMP Specification Overview) BAP-C1605-005-A." For details of the contents described in the table, see the "PDU List" in the "CC-Link IE Field Network Specification (SLMP Specification Overview) BAP-C1605-005-A."] ASCII and Binary indicate that the data is to be sent in ASCII mode or Binary mode, respectively. If a response is not performed (not required), omit this item.

[Description example] rdReqST_Binary, wrReqMT_ASCII

Table 4-19 PDU Name List for SLMP Response

Type	SLMPPDU	PDU Name
ST	rdResST-PDU	rdResST
	wrResST-PDU	wrResST
MT	rdResMT-PDU	rdResMT
	wrResMT-PDU	wrResMT

4.3.1.41 Convention of RESULTx

Write RESULTx in order to show the value applied when conditions are satisfied in the conditionMember division of CONDITION part (See Section 5.4.9). See Section 5.4.9 for details of this convention.

(1) A maximum of 32 RESULTx items can be written in 1 conditionMember division.

4.3.1.42 TARGET conventions

A TARGET item describes a reference to an element of a COMM_IF_INPUT part, COMM_IF_OUTPUT part, COMM_IF_PARAMETER part, or COMM_IF_COMMAND part used in a method or message. The conventions are indicated below. For the specifications of other items in a case where a reference is described in the TARGET item, see Section 4.3.4.

(1) In a case where reference is to be made to multiple elements in a TARGET item, write the description based on one of the below. Note, however, that you cannot write multiple references to an element of a COMM_IF_COMMAND part.
Further, you cannot write a reference that mixes elements of different part types.
[1] List the elements as a set.
[2] Describe the elements with a "part label name.*" (when reference is to be made to all elements included in one part).
(Example: CommParameter.*)
[3] Use a combination of the above (when reference is to be made to all elements of multiple parts).

(2) Write "PARENT_TARGET" in a TARGET item of a method that is to serve as reference by describing the RELATED_METHOD item.

(3) Write "SEQ_TARGET" in elements that use SEQ in a RELATED_METHOD (see Section 4.3.1.33) and RELATED_MESSAGE (see Section 4.3.1.32). Communication information cannot be described in elements where "SEQ_TARGET" is written. Thus, the following items cannot be described.

Table 4-20 Items that Cannot Be Written in Elements WhereSEQ_TARGET Is Written

Part Type	Items that Cannot Be Written
Method	WRITE_REGISTER, WRITE_DATA, WRITE_DATATYPE, READ_REGISTER, READ_DATA, READ_DATATYPE, INTERLOCK, REQ_FLAG, END_CONDITION, ERR_CONDITION, ERR_REGISTER
Message	REQUEST_TYPE, REQUEST_DATA, REQUEST_DATATYPE, RESPONSE_TYPE, RESPONSE_DATA, RESPONSE_DATATYPE, ERR_TYPE, ERR_CODE_RANGE

4.3.1.43 UI_ATTRIBUTE conventions

A UI_ATTRIBUTE item describes the display window number and display sequence of elements of a utility software display. The item describes the two values shown in Table 4-21 in order, within brackets. This item may be omitted if not required.

When you want to display one element in multiple utility software windows, write multiple numbers as a set (see Section 4.3.3, [2]).

(Example: [<1><1>] [<2><1>]

With an XML file: [<item><1><1></item><item><2><1></item>])

Table 4-21 List of Input Items of Common Utility Software Display Information

No.	Name	Meaning	Setting value
1.	Window number (*1)	Describes the window number to be displayed on the tool. ・ Elements in which the same number is set are set up and monitored in the same window. Even if the elements are from different BLOCK sections, the elements appear in the same window when the same window number is written. ・ The name of each window is written in the element "UI_ATTRIBUTE_Window**" of the DEVICE_INFO part. Mapping is performed using the value of "**" and the window number. A "UI_ATTRIBUTE_Window**" that corresponds to a window number is not required (the item does not have to be written). ・ When the UI_ATTRIBUTE item is written, the window number cannot be omitted.	UINT8
2.	Display sequence	Describes the display sequence for the elements to be displayed in the same window. ・ The elements are displayed in ascending order on each window. ・ The display sequence when the same value is set is freely determined on the utility software side. ・ Specify the window number and enter "-" when specification of a display sequence is not required (example: <6><->). In this case, the display sequence is freely determined on the utility software side. ・ The "display sequence" is the display sequence, not the number of rows displayed. The same value can be written in another element, as described above. For utility software that displays structure types in multiple rows, even without value duplication, the number of rows is no longer indicated if the structure types are mixed.	UINT16, -

*1: While the name is "Window number," this does not necessarily mean that the elements are divided into windows when displayed using utility software. For example, a display in which the elements are divided and switched via tabs, etc., is unproblematic.

* Elements can be classified using the UI_ATTRIBUTE item as well as the CATEGORY item. In a case where it is best to divide the screen, classify the elements using the UI_ATTRIBUTE item. To classify elements within a screen, we recommend using the CATEGORY item.

4.3.1.44 WRITE_DATA conventions

A WRITE_DATA item indicates the value to be written and is described following the conventions below. Note that the value specified needs to comply with the data type of the WRITE_DATATYPE item (see Section 4.3.1.45).

(1) Write the value directly. (Example: 0x40)

(2) Specify the value by writing a reference to the VALUE of an element of a TARGET item (see Section 4.3.1.42), the size of a DATATYPE item (see Section 4.3.1.10), an ASSIGN item (see Section 4.3.1.3), and an element of a COMMAND_ARGUMENT part, placing the reference between "$(" and ")" (see Section 4.3.3).
(Example: $(VALUE), <$(ARGUMENT.WriteFrequencyData.VALUE)>, $(DATATYPE).WORDSIZE)
* When specifying the element of COMMAND_ARGUMENT, you cannot specify an element that is not readable (NA, R, etc.) by the ACCESS item (see Section 4.3.1.1).
* When describing the writing of all elements inside TARGET using an asterisk (example: $(*.VALUE)), you cannot describe an element that is not writable in TARGET.
* When multiple elements are referenced in TARGET and one of the elements is to be selected and written, you can write an element that is not writable in TARGET.
* In an element in which the METHOD_TYPE is PARAMETER and VALUE is referenced by READ_DATA, you cannot describe a VALUE reference in WRITE_DATA.

(3) When multiple write registers have been described using the WRITE_REGISTER item (see Section 4.3.1.47), specify the same number of write values as write registers, placing each WRITE_DATA item between "<" and ">" (see Section 4.3.3) as well.
(Example: <0x40><$(VALUE)>, <$(*.VALUE)>)

(4) This item cannot be written when the WRITE_REGISTER item is omitted.

(5) When processing that specifies and writes elements of a STRUCT type, which has empty space in the offset, to a WRITE_DATA item of a method, zero is written to the empty section of the offset. When a fixed value other than zero needs to be written, write an element in which the DATATYPE is constant inside the STRUCT part.

4.3.1.45 WRITE_DATATYPE conventions

A WRITE_DATATYPE item describes the data type of the data to be written, using the conventions below.

(1) Write the data type directly (see Section 4.3.1.10).

(2) Specify the data type by writing a reference to a DATATYPE item (see Section 4.3.1.10) of an element, placing the reference between "$(" and ")" (see Section 4.3.3).
(Example: $(DATATYPE))

(3) When multiple WRITE_DATA items (see Section 4.3.1.44) have been written using the WRITE_REGISTER item (see Section 4.3.1.47), specify the same number of data types as WRITE_DATA items, placing each WRITE_DATATYPE item between "<" and ">" (see Section 4.3.3) as well.
(Example: <WORD><$(DATATYPE)>, <$(*.DATATYPE)>)

(4) This item cannot be written when the WRITE_REGISTER item is omitted.

4.3.1.46 WRITE_ORDER conventions

A WRITE_ORDER item sets the write order when the utility software writes multiple parameters to a module. The write order specifications are described below.

(1) The writable values are 0 to 65535 (in decimal format).

(2) Write the numerical values starting from the lower number.

(3) Duplication of numerical values is permitted. If there is a duplicate, the order in which the parameters are written is freely determined on the utility software side.

(4) The numerical values do not need to be continuous. Numerical value gaps are permitted.

(5) The order does not need to be specified. If the element can be written last, this item may be omitted.

4.3.1.47 WRITE_REGISTER conventions

A WRITE_REGISTER item specifies the register for writing following the conventions below. The value written needs to comply with the WRITE_DATA item (see Section 4.3.1.44).

(1) Write the link register directly using a string. For conventions, see the ASSIGN item (see Section 4.3.1.3). (Example: RWw1)

(2) Specify the register by referring to the ASSIGN item of the element using a reference specification based on "$(ASSIGN)" when a reference has been written in the TARGET item (see Section 4.3.1.42). (Reference cannot be made to any item other than the ASSIGN item.)
(Example: $(InitParameters.CH1_AvgTime.ASSIGN), $(ARGUMENT.WriteFrequencyData.ASSIGN))
* Reference cannot be made to a LABEL item (see Section 4.3.1.19) of an element not described in the TARGET item.
* An element or argument in which the ACCESS item (see Section 4.3.1.1) is not writable ("NA," "R," etc.) cannot be specified. Note, however, that specification is unproblematic in a case where a description that refers to multiple elements in the TARGET item is written and the elements include an element that is not writable.

(3) If you want to specify multiple registers for writing, write the registers in list format in order, placing each register name between "<" and ">" (see Section 4.3.3.).
(Example: <RWw4><$(ASSIGN)>)
Note that when multiple registers are written, writing is performed in order, starting from the register written on the left.

(4) In a case where there are multiple content descriptions in the corresponding WRITE_DATA item, including content that does not need to be written, write the content description of the WRITE_REGISTER item corresponding to the content that does not need to be written as $(DUMMY). The content of the WRITE_DATA item corresponding to $(DUMMY) will not be read written to the register. (Example: In a case where three content descriptions exist in the WRITE_DATA item and the first content does not need to be written, write the WRITE_REGISTER item as follows: <$(DUMMY)><RWw2><RWw3>.)

(5) This item may be omitted if the method does not require writing.

When describing the WRITE_REGISTER item, describe the item so that writing is performed in units of 1 bit or in units of words. Note that if you specify writing in units of words by, for example, describing writing in pairs to RWw0.0 with a data type of UINT8 and writing to RWw0.8 with a data type of UINT8, in the WRITE_REGISTER item of a single METHOD part element, each WRITE_REGISTER item does not need to be set to word units.

4.3.2 Prohibited XML character support

If you want to use prohibited XML characters as text between tags (<Tag> Use here </Tag>), write the characters in XML using an entity reference as shown in Table 4-22.

Table 4-22 Entity References for Prohibited Characters

Prohibited Character	Entity Reference
<	<
>	>
&	&

* While entity references are defined for a double quote (") and apostrophe (') as well, these characters can be used as is without using the entity references in XML text sections (use entity references only when using the characters with attribute values). With CSP+ attribute values, these characters cannot be written. Therefore, be sure to use the " and ' entity values with CSP+.

4.3.3 Common conventions related to item values

The following describes the common description rules that apply when writing item values.

Table 4-23 Description Rules

No.	Rule	Meaning	Remarks
[1]	"<>" (single-byte angle brackets)	When you want to describe multiple contents in an item, describe the multiple contents in list format in order, placing each between "<" and ">" (single-byte angle brackets.) When you use single-byte angle brackets, you need to use the brackets with all values of that cell. Sets and single-byte angle brackets cannot be used in locations enclosed by single-byte angle brackets. [Examples] "<0x04><70>" and "'<0x16><70>' '<0x18><80>'" are acceptable. "<0x04>70", "'>0x16><70>' '0x18'", and "<<0><1><2>>" are not acceptable. When a reference is used such as "<0x04>$(aaa.bbb.ASSIGN)," the value of the reference is acceptable if enclosed in angle brackets. * " 'A' 'B' " indicates a set of A and B.	"<" ">" (single-byte angle brackets) fall under the category of prohibited XML characters. In CSP+, write the brackets as "<" and ">". (See Section 4.3.2.) [Description Example] <assign> <item> <0x04><70><3> </item> </assign>
[2]	Description of multiple data as a set	When you want to describe multiple data for the same item in a single cell, write "item" multiple times in list format.	[Description Example] <ref> <item> CH1.BlockParameter.OffsetGain </item> <item> CH2.BlockParameter.OffsetGain </item> </ref>
[3]	"$()"	Use this when you want to differentiate a reference description from a regular string (such as for a METHOD argument, etc.). Enter round parentheses after "$", writing the reference inside the round parentheses. For the reference description method, see Section 4.3.4. * A reference is always described with REF (see Section 4.3.1.30) and TARGET (see Section 4.3.1.42). In these cases, directly write the reference without using "$()".	[Description Example] $(InitParameters.OffsetGain)
[4]	Value range specification "[", "(", ")", "]"	A value range is indicated by entering two values separated by a comma (",") between brackets. Ranges can only be specified using numerical values, not strings. The left value indicates the lower limit, and the right value indicates the upper limit. A closed interval is expressed using "[" , "]" (single-byte square brackets), and an open interval is expressed using "(", ")" (single-byte round brackets). In a case where the upper limit or lower limit is not to be specified, do not write a numerical value. (Example: For a value less than or equal to 6, write [,6].) Further, both the upper limit and the lower limit may be omitted. (The range then becomes that indicated by the DATATYPE item.)	• Closed interval [a,b] = {x | a ≤ x ≤ b} • Open interval (a,b) = {x | a < x < b} • Closed left and open right interval, closed left semi-open interval [a,b) = {x | a ≤ x < b} • Open left and closed right interval, closed right semi-open interval (a,b] = {x | a < x ≤ b}

No.	Rule	Meaning	Remarks
[5]	"*"	[METHOD and MESSAGE limit conventions] When you want to refer to all elements included in a single part, write "Part label name.*" or "<$(*.Item name)>". For details, see Section 4.3.4.	[Description Example] CommParameter.* <$(*.VALUE)>
[6]	Treatment of special characters	Excluding items in which any string is writable, when you want to use any of the following symbols in a string as a character, write the code symbol following "¥". "$" (dollar) "<", ">", "[", "]", "(", ")" (brackets) "¥" (backslash) [Items in which any string is writable] * While the convention "$()" of (3) is applicable with a DATA item of designated elements, the convention cannot be used with any of the other items. For this reason, special characters can be written without using "¥".	[Description Example] ¥, ¥$, ¥<, ¥>, ¥[, ¥], ¥(, ¥), ¥¥
[7]	Space input	You can freely use spaces with any item that permits input of any string and with any cell that treats input values as a string type. However, you cannot use spaces in cells that treat input values as any value other than a string type.	
[8]	Treatment of line feeds	To make strings easier to view when displayed using utility software, you can enter a line feed in NAME, COMMENT, and DATA items only. You cannot insert a line feed in any other item. Note, however, that whether or not the line feed code is reflected at the time of display depends on the specifications of the utility software (not required).	
[9]	Use of common information • Structure (STRUCT) • Option list (ENUM) • Command argument list (COMMAND_ARGUMENT)	Reference can be made to common information defined in parts.	
[10]	"#"	Use this when you want to specify up to the elements defined within the STRUCT part when referring to an element in which the data type is structure. (See Section 4.3.4.)	This may be combined with any convention. For example, you can write # within $(). Note, however, that you cannot write "#" in multiple within a single reference description. (The reference description AA.BB#CC.DD#EE is not possible.)
[11]	Handling of control characters	Control characters other than return cannot be written.	Example: TAB (horizontal tab) cannot be written.

Table 4-24 List of Conventions Applicable to Each Item

		Multiple Descriptions Using <>[(1)]	Multiple Descriptions as a Set [(2)]	Reference Description Using $() [(3)]	Range Specification Using [] or () [(4)]	Reference to All Elements Using * [(5)]	STRUCT Reference (*1)	ENUM Reference	COMMAND_ARGUMENT Reference	CONDITION Reference (*3)
1	LABEL	X	X	X	X	X	X	X	X	X
2	LABEL2	X	X	X	X	X	X	X	X	X
3	CATEGORY	X	X	X	X	X	X	X	X	X
4	NAME	X	X	X	X	X	X	X	X	X
5	DATATYPE	X	X	X	X	X	O	X	X	X
6	DEFAULT	X	X	X	X	X	X	X	X	X
7	RANGE	X	O	X	O	X	X	O	X	O
8	MIN_INC	X	X	X	X	X	X	X	X	O
9	ENG_UNIT	X	X	X	X	X	X	X	X	O
10	ASSIGN	O	X	X	X	X	X	X	X	X
11	ACCESS	X	X	X	X	X	X	X	X	O
12	REF	X	O	X	X	X	X	X	X	X
13	COMMENT	X	X	X	X	X	X	X	X	X
14	UI_ATTRIBUTE	O	O	X	X	X	X	X	X	X
15	WRITE_ORDER	X	X	X	X	X	X	X	X	X
16	DATA	*6	O	*2	X	X	X	X	X	X
17	ARGUMENT	X	X	X	X	X	X	X	O	X
18	TARGET	X	O	X	X	O	X	X	X	X
19	METHOD_TYPE	X	O	X	X	X	X	X	X	X
20	WRITE_REGISTER	O	X	O	X	O	X	X	X	X
21	WRITE_DATA	O	X	O	X	O	X	X	X	X
22	WRITE_DATATYPE	O	X	O	X	O	X *4	X	X	X
23	READ_REGISTER	O	X	O	X	O	X	X	X	X
24	READ_DATA	O	X	O	X	O	X	X	X	X
25	READ_DATATYPE	O	X	O	X	O	X *4	X	X	X
26	INTERLOCK	X	X	O	X	X	X	X	X	X
27	REQ_FLAG	X	O	O	X	X	X	X	X	X
28	END_CONDITION	X	O	O	X	X	X	X	X	X
29	ERR_CONDITION	X	X	O	X	X	X	X	X	X
30	ERR_REGISTER	X	X	O	X	X	X	X	X	X
31	RELATED_METHOD	O	X	X	X	X	X	X	X	X
32	MESSAGE_TYPE	X	O	X	X	X	X	X	X	X
33	REQUEST_TYPE	O	X	X	X	X	X	X	X	X
34	REQUEST_DATA	O	X	O	X	O	X	X	X	X
35	REQUEST_DATATYPE	O	X	O	X	O	X *4	X	X	X
36	RESPONSE_TYPE	O	X	X	X	X	X	X	X	X
37	RESPONSE_DATA	O	X	O	X	O	X	X	X	X
38	RESPONSE_DATATYPE	O	X	O	X	O	X *4	X	X	X
39	ERR_TYPE	O	X	X	X	X	X	X	X	X
40	ERR_CODE_RANGE	X	O	X	O	X	X	O	X	X
41	RELATED_MESSAGE	O	X	X	X	X	X	X	X	X
42	OFFSET	X	X	X	X	X	X	X	X	X
43	CODE	X	X	X	X	X	X	X	X	X
44	CONDITIONXVARIABLE	X	X	X	X	X	X	X	X	X
45	CONDITIONx	X	O	O	O	X	X	X	X	X
46	RESULTx	X	*5	*5	*5	X	X	*5	X	X
47	RELATED_ELE *7	X	X	X	X	X	X	X	O	X

*1: For STRUCT part references, applicable items as well as applicable parts are predetermined. (See Section 4.3.1.10.)
*2: $() of a DATA item can only be written with specific elements defined in the specification.
*3: Can only be written with elements of BLOCK_PARAMETER.
*4: Cannot write direct references, but it does not matter if the DATATYPE item of the elements referenced with the TARGET item is STRUCT type. Therefore, the result of solving the reference of $(DATATYPE) can be of STRUCT type.
*5: Follows the specifications of the item which is the reference source of RESULTx. (Ex: range specifications with [] or () can be used in RESULTx referenced from RANGE items.)
*6: Brackets in DATA can only be written with specific elements of CC-Link IE TSN (the elements in which the data type is specified as the array type).
*7: Can only be used under the specific conditions. For details, see descriptions of each item.

The following shows the combinability of these conventions in a single item. The combinability in relation to common information (STRUCT part, ENUM part, and COMMAND_ARGUMENT part) references is described below.

Table 4-25 shows the combination rules related to other conventions.

(1) STRUCT part
In an item that describes a reference to a STRUCT part, combined use with other conventions is not permitted.

(2) ENUM part
In an item that describes a reference to an ENUM part, combined use with other conventions is not permitted .

(3) COMMAND_ARGUMENT part
In an item that describes a reference to a COMMAND_ARGUMENT part, combined use with other conventions is not permitted .

(4) CONDITION parts
In items which have written references to CONDITION parts, this convention cannot be used with other conventions.

Table 4-25 Convention Combined Use Rules

	"-" (Abbreviation some values)	"< >" (Multiple value representation by half-width angle brackets)	"," (Multiple value representation by commas)	"$()" (Reference)	"[", " (", ")", "]" (Specified range of values)	"*" (wild card)
-	Multiple arrangements are prohibited. However, multiple descriptions of <-> are permitted. (Ex: <3><->, <5><-> in UI_ATTRIBUTE is permitted)					
<>	- can be written inside < >.	Multiple description such as "<a><b>" are permitted. Writing <> inside <> is not permitted.				
Multiple Representations	- is only permitted when inside < >.	Description like "<3><4>, <5><6>" is permitted. <,> are not permitted.	Multiple description such as "a, b, c" is permitted.			
$()	Combination not permitted	Combination such as "$()<a><b>" are permitted. <$()> are also permitted. $(<>) is not permitted.	Combination such as $(), $() is permitted. $(,) is not permitted.	$($()) can be used. Combination such as $(),$() is permitted.		
"[", "(" , ")", "]"	Combination not permitted	Combination not permitted	Combination is permitted when writing multiple ranges such as " [1, 2], [10, 15]". Only one "," can be written inside brackets.	Combination is permitted. (Combination when specifying upper limit and lower with set values of other parameters)	Multiple description using commas is permitted, such as "[1, 2], [10, 15]". Writing brackets inside brackets is not permitted.	
*	Combination not permitted	Combination such as "<$(*.VALUE>)" is permitted.	Multiple description of reference using * is permitted.	Combination permitted such as "<$(*.VALUE>)".	Combination not permitted	Multiple description using commas is permitted, such as "xx.*, yy.*". Writing multiple "*" in strings not separated by commas or < > is not permitted, such as "*.xx.*".

×: Combined use not permitted

4.3.4 Conventions for references that use labels

(1) Common conventions
When you want to refer to a value of a part, element, or item of an element, you can describe the reference using a label name (you cannot use a Label2 name). The following describes the description specifications for references based on a label name.

(a) When you want to refer to an element within the same part, write only the label name of the element.
(b) When you want to refer to a part within the same section, write the label name of the part.
(c) When you want to refer to an element of a different part within the same section, write "Part label name. Element label name".
(d) When you want to refer to a value of an item of an element, write "Element label name.Item name". Note, however, that the specifiable items are only ASSIGN, DATATYPE, and ARGUMENT items.
(e) When you want to refer to an element beyond the section, write "Section label name.Part label name. Element label name", adding the label name of the section at the start of the path. Note, however, that for common information (STRUCT part, ENUM part, COMMAND_ARGUMENT part), a reference cannot be made beyond the section.
(f) The type of reference (part, element, item) is uniquely determined by the item that describes the reference, as shown in Table 4-26. When determining a reference, make sure that the reference type is appropriate.
(g) When the data type of the element of the reference is structure and you want to refer to only one element defined within the STRUCT part, use # and write "Element label name of reference#Element label name within STRUCT part".
(h) When the data type of the element of the reference is structure and you want to refer to the address information of one element defined within the STRUCT part, use # and write "Element label name of reference#Element label name within STRUCT part.ASSIGN".
* While STRUCT does not have the item ASSIGN, reference is made to the ASSIGN item to indicate that reference is being made to address information, making a distinction from a case where reference is made to the value of the item OFFSET itself.

Table 4-26 List of Reference Writable Items

Reference	Reference Writable Item	Remarks
Reference to part	Items in which common information (STRUCT, ENUM, COMMAND_ARGUMENT) can be used	While references to elements and references to items overlap in terms of writable items, the conventions for common information are predetermined, and thus the reference is assessable. For items that can use common information, see Table 4-24.
Reference to element	REF, TARGET, RELATED_METHOD, RELATED_MESSAGE	-
Reference to item	Items in which a reference description using $() can be used in Table 4-24 • Reference to RESULTx item of CONDITION part • Reference to item used for conditional determination from CONDITIONXVARIABLE item of CONDITION part.	-

(2) Conventions applicable to items within METHOD and MESSAGE parts only

[1] When you want to refer to all elements included in a single part in a TARGET item, write "Part label name.*".
Note that "*" is applicable to references to elements only. You cannot write "Section label name.*" to refer to all parts within a section.

[2] If you want to refer to an item of an element specified by a TARGET item set up in a METHOD or MESSAGE part from an item other than a TARGET item, omit the path up to the element of the part and write the item name only.

[3] If "<$(*.Item name)>" is written, the description describes all elements of the TARGET item enclosed by angle brackets based on the conventions of (1) of Table 4-23.
(Example: "<$(*.ASSIGN)>" is equivalent to "<$(First element of TARGET.ASSIGN)> <$(Second element of TARGET.ASSIGN)><(Third element of TARGET.ASSIGN)>. . . ".)

[4] In the TARGET item of a METHOD part referred to by a RELATED_METHOD item (see Section 4.3.1.33), be sure to write "PARENT_TARGET", which indicates that the same element as the TARGET item used in the reference source is always used. The same holds true for the TARGET item of a MESSAGE part referred to by a RELATED_MESSAGE item (see Section 4.3.1.32).

[5] If you want to describe a reference to argument information referred to by an ARGUMENT item of a COMM_IF_COMMAND part in a method or message that refers to the element of a COMM_IF_COMMAND part in a TARGET item, enter a "." after "ARGUMENT" followed by a description of the reference to the element of the argument.
(Example: <$(ARGUMENT.WriteFrequencyData.ASSIGN)>)

[6] When you want to write or read all arguments of a command when the TARGET item is a command, write <$(ARGUMENT.*.Item name)>. ("VALUE" can be written as the item name as well.)
Example: In a case where the COMMAND_ARGUMENT of a COMMAND part referred to by TARGET has the elements "Ele1", "Ele2", and "Ele3", then <$(ARGUMENT.*.VALUE)> written in the WRITE_DATA item means
<$(ARGUMENT.Ele1.VALUE)><$(ARGUMENT.Ele2.VALUE)><$(ARGUMENT. "Ele3.VALUE)>, for example.

[7] When you want to acquire a write value from utility software for inputs, outputs, parameters, and commands specified in a TARGET item, or when you want to indicate that a read value is to be displayed with utility software, use VALUE as described below.
- For inputs, outputs, parameters: $(VALUE)
- For COMMAND arguments: $(ARGUMENT.Element label name.VALUE)

Such a description indicates that the value acquired from the utility software is to be used when written in the WRITE_DATA or REQUEST_DATA item, or that the read value is to be returned to the utility software when written in the READ_DATA or RESPONSE_DATA item.
Note that VALUE based references can only be used with references to elements specified by the TARGET item, as described above. Such references cannot be used as references to regular elements. (A description such as Section LABEL.PartLABEL.Element LABEL.VALUE is not permitted.)

[8] When you want to indicate that the data size of an element selected in the TARGET item is to be written using a REQUEST_DATA item or WRITE_DATA item, write the description using the description method of

Table 4-27. Note that it is not possible to write the data size by directly referring to an element other than that of TARGET. [A description such as $(CommParameter.SelectParam.DATATYPE.WORDSIZE) is not permitted.]

* For example, depending on the SLMP command, the data size needs to be written in the REQUEST_DATA item. When multiple parameters are specified in the TARGET item and the data size differs for each parameter, you can switch the data size for each parameter selected by using this convention.

Table 4-27 Description Method of Data Size of Element

No.	Description Method	Meaning
1.	$(DATATYPE).WORDSIZE	Indicates the word size of the data type of the referred element. Indicates 2 for DWORD and 4 for LREAL, for example. Applicable to structure-type elements as well. Indicates a raised value if the latter half of the word is not assigned. (Example: Indicates 3 if only up to 2.A is used within STRUCT.)
2.	$(DATATYPE).BITSIZE	Indicates the bit size of the data type of the referred element. Indicates 1 for BOOL and 8 for BYTE, for example. Applicable to structure-type elements as well. Does not raise a value as is done with WORDSIZE, but rather indicates the used value as is. (Example: Indicates 42 if only up to 2.A is used within STRUCT.)

[9] Reference from each item of a METHOD or MESSAGE part can only be made to a part or element within the same COMM_IF section. Reference cannot be made directly to an element of a BLOCK section. (Refer to an element of a COMM_IF section that refers to a BLOCK section from a METHOD or MESSAGE part.)

5. CSP+ File Details

This chapter describes the details of CSP+ file conventions.

A single CSP+ file is written as a single XML element enclosed by the XML tag name "profile" (name space: "http://cc-link.org/cspplus/ver1/"; hereinafter the same and thus omitted). The following shows a description example of an XML tag that expresses a single CSP+ file description.

```
<p:profile xsi:schemaLocation="http://cc-link.org/cspplus/ver1/ CSPPSchema.xsd"
   xmlns:p="http://cc-link.org/cspplus/ver1/"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      :
      :
      :
</p:profile>
```

[Name Space]
With XML, a name space is used to indicate the specifications in which the XML element name or attribute name is defined. In the above example,

http://cc-link.org/cspplus/ver1/
http://www.w3.org/2001/XMLSchema-instance

written in lines two and three indicate the name space, with the former serving as the name space of CSP+ specifications, and the latter as the name space of XML specifications.
The name space increases in length when described for all XML element names and attribute names, and thus an abbreviated format is declared and used as is done in lines two and three in the above example.
The abbreviated format of the name space is declared using the following format:

xmlns: Prefex = "Name space"

In this example, the prefix "p: " expresses the name space of CSP+ specifications on line two, and the prefix "xsi" expresses the name space of XML specifications on line three. In the above example, p:profile is the XML element name defined in CSP+ specifications, and xsi:schemaLocation is the attribute name defined in XML specifications. The string used in the abbreviated format may be a string other than that shown in the above example as well. For example, in a case where the following is declared in line two:

"xmlns:cspp="http://cc-link.org/cspplus/ver1/"

the abbreviated format is written as "cspp:profile" rather than "p:profile".
With CSP+ descriptions, the attribute defined in CSP+ specifications is described "without a name space".

Table 5-1 shows a list of CSP+ XML elements, and Table 5-2 shows a list of attributes of the element "profile".

Table 5-1 CSP+ XML Element List

No.	XML Element Name	Description Details	Required/Optional
1.	profile	Describes CC-Link family protocol (CSP+).	Required

Table 5-2 profile Element Attribute List

No.	Attribute Name	Description Details	Required/Optional
1.	schemaLocation	Indicates the location of XML schema of CSP+. This attribute is defined in XML specifications, and thus is used along with the name space "http://www.w3.org/2001/SMLSchema-instance" defined in XML specifications. Specify "http://cc-link.org/cspplus/ver1/CSPPSchema.xsd".	Optional

Further, an element that expresses a section is written as one XML child element of the profile element indicated by CSP+. One section is described as one XML element. Table 5-3 shows a list of child elements of the profile element.

Table 5-3 List of XML Child Elements of profile Element

No.	XML Element Name	Description Details	Required/Optional
1.	file	Describes the contents of the FILE section	Required
2.	device	Describes the contents of the DEVICE section	Required
3.	block	Describes the contents of the BLOCK section	Required
4.	commIf	Describes the contents of the COMM_IF section	Required

5.1 FILE Section

The FILE section comprises one FILE_INFO part only. The one part is written as a single XML element. A comment may also be written (option).

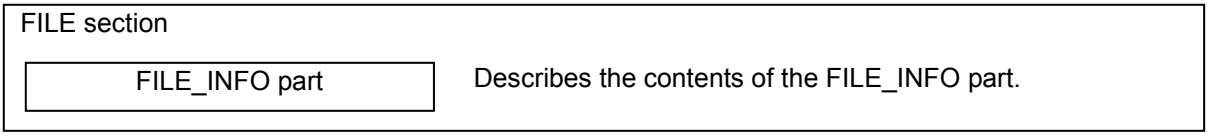


Figure 5-1 Structure of FILE Section

In XML format, the FILE section is written as an XML element of the XML element name "file". Table 5-4 shows a list of the attributes of the file element, and Table 5-5 shows a list of XML child elements.

Table 5-4 List of Attributes of file Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label that identifies the FILE section. (See Section 4.3.1.19.) (Example: "FileSection")	Required

Table 5-5 List of XML Child Elements of file Element

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the FILE section. (See Section 4.3.1.6.)	Optional
2.	fileInfo	Describes the contents of the FILE_INFO part.	Required

5.1.1 FILE_INFO part

5.1.1.1 Overview

A FILE_INFO part describes information related to the CSP+ file, such as the file modification date and time.

Figure 5-2 shows the elements that make up the FILE_INFO part.

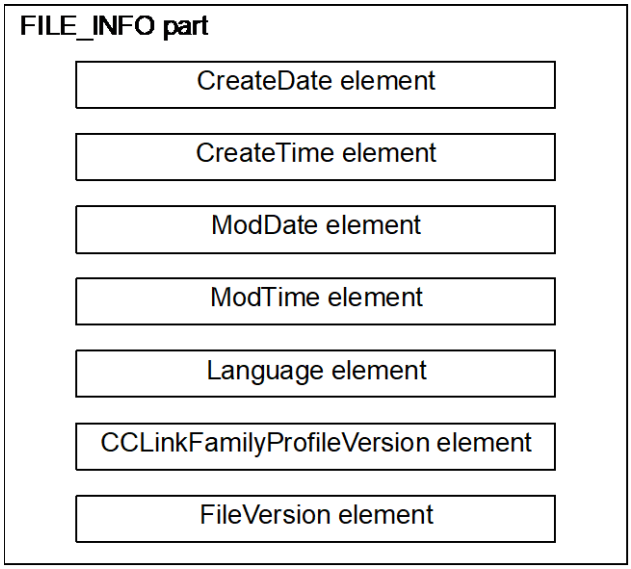


Figure 5-2 Structure of FILE_INFO Part

The configuration of each element of the FILE_INFO part, that is, the items that should be described within each element, is the same. Figure 5-3 shows the configuration of an element of the FILE_INFO part.

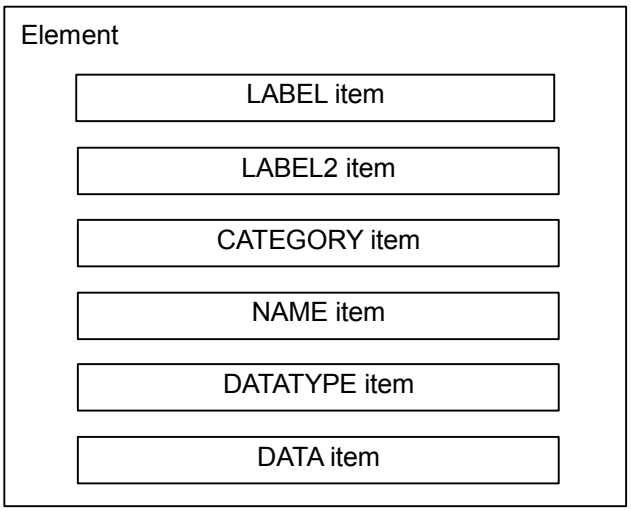


Figure 5-3 Structure of Element of FILE_INFO Part

Table 5-6 lists the elements. A single element is described as a single XML element. For the detailed description specifications of each element, see Section 5.1.1.3.

Table 5-6 Element List for FILE_INFO Part

No.	Element Name	Description Details	Required/Optional
1.	CreateDate	Describes the date on which the CSP+ file was created.	Required
2.	CreateTime	Describes the time at which the CSP+ file was created. (Used when the utility software supports another language.)	Required
3.	ModDate	Describes the last modification date.	Required
4.	ModTime	Describes the last modification time.	Required
5.	Language	Describes the information of the language in which the CSP+ file is written.	Required
6.	CCLinkFamilyProfileVersion	Describes the version of CSP+ description specifications.	Required
7.	FileVersion	Describes the version of the profile information for the target equipment.	Required

Table 5-7 shows the items described in the elements of the FILE_INFO part.

Table 5-7 Item List for FILE_INFO Part

No.	Item Name	Description Details	Description Specifications	Required /Optional
1.	LABEL	Describes the label for identifying the element.	See Section 4.3.1.17	Required
2.	LABEL2	Describes the label for identifying the element. (Used when the utility software supports another language.)	See Section 4.3.1.18	Optional
3.	CATEGORY	Describes the category for grouping the element.	See Section 4.3.1.4	Optional
4.	NAME	Describes the name of the element.	See Section 4.3.1.22	Optional
5.	DATATYPE	Describes the data type of the contents described in the DATA item.	See Section 4.3.1.8	Required
6.	DATA	Describes the contents of the element.	See Section 4.3.1.7	Required

5.1.1.2 XML format of FILE_INFO part

In XML format, the FILE_INFO part is described as an XML element having the XML element name "fileInfo". Table 5-8 shows a list of the attributes of the fileInfo element, and Table 5-9 shows a list of the XML child elements.

Table 5-8 Attribute List for fileInfo Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the FILE_INFO part. (See Section 4.3.1.19.) (Example: "FileInformation")	Required

Table 5-9 XML Child Element List for fileInfo Element

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the FILE_INFO part. (See Section 4.3.1.6.)	Optional
2.	fileInfoMember	Describes the contents of the elements of the FILE_INFO part.	Required

In XML format, the element of the FILE_INFO part is described as an XML element having the XML element name "fileInfoMember". Table 5-10 shows a list of attributes of the fileInfoMember element, and Table 5-11 shows a list of XML child elements.

Table 5-10 Attribute List for fileInfoMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element.(See Section 4.3.1.19.) The contents written for this label are regulated by CSP+ specifications. For details, see Section 5.1.1.3.	Required

Table 5-11 XML Child Element List of fileInfoMember Element

No.	XML Element Name	Description Details
1.	label2	Describes the LABEL2 item
2.	category	Describes the CATEGORY item
3.	name	Describes the NAME item
4.	datatype	Describes the DATATYPE item
5.	data	Describes the DATA item

Write the FILE_INFO part using a format such as shown below.

```
<p:fileInfo label="Write in accordance with the description details of Table 5-8 'Attribute
Name label'''>
   <p:comment>''Write in accordance with the description details of Table 5-9 'XML Element
   Name comment'''</p:comment>
   <p:fileInfoMember label=''Write in accordance with the description details of Table 5-10
   'Attribute Name label'''>
      <p:label2>''Write in accordance with the description details of Table 5-11 'XML Element
      Name label2'''</p:label2>
      <p:category>''Write in accordance with the description details of Table 5-11 'XML Element
      Name category'''</p:category>
      <p:name>''Write in accordance with the description details of Table 5-11 'XML Element
      Name name'''</p:name>
      <p:datatype>''Write in accordance with the description details of Table 5-11 'XML Element
      Name datatype'''</p:datatype>
      <p:data>''Write in accordance with the description details of Table 5-11 'XML Element
      Name data'''</p:data>
   </p:fileInfoMember>
   <p:fileInfoMember label=''Write in accordance with the description details of Table 5-10
   'Attribute Name label'''>
      :
      :
   </p:fileInfoMember >
   :
   :
</p:fileInfo>
```

5.1.1.3 Description details of each element

(1) Item description of CreateDate element

Table 5-12 Item Description Specifications of CreateDate Element

No.	Item Name	Description Details
1.	LABEL	Write "CreateDate"
2.	LABEL2	Write "CreateDate" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "File Creation Date" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(10)"
6.	DATA	Write the file creation date. Write the date in the format yyyy/mm/dd. Write the year based on the Western calendar. Do not suppress any zeros in the month or day.

(a) Item description example of CreateDate element

```
<p:fileInfoMember label="CreateDate">
   <p:label2><p:item>CreateDate</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>File Creation Date</p:item></p:name>
   <p:datatype><p:item>STRING(10)</p:item></p:datatype>
   <p:data><p:item>2012/04/01</p:item></p:data>
</p:fileInfoMember>
```

(2) Item description of CreateTime element

Table 5-13 Item Description Specifications of CreateTime Element

No.	Item Name	Description Details
1.	LABEL	Write "CreateTime"
2.	LABEL2	Write "CreateTime" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "File Creation Time" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(8)"
6.	DATA	Write the file creation time. Write the time in the format hh:mm:ss. Do not suppress any zeros in the hour, minute, or second.

(a) Item description example of CreateTime element

```
<p:fileInfoMember label="CreateTime">
   <p:label2><p:item>CreateTime</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>File Creation Time</p:item></p:name>
   <p:datatype><p:item>STRING(8)</p:item></p:datatype>
   <p:data><p:item>09:08:07</p:item></p:data>
</p:fileInfoMember>
```

(3) Item description of ModDate element

Table 5-14 Item Description Specifications of ModDate Element

No.	Item Name	Description Details
1.	LABEL	Write "ModDate"
2.	LABEL2	Write "ModDate" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Last Modified Date" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(10)"
6.	DATA	Write the last modification date. Write the date in the format yyyy/mm/dd. Write the year based on the Western calendar. Do not suppress any zeros in the month or day.

(a) Item description example of the ModDate element

```
<p:fileInfoMember label="ModDate">
   <p:label2><p:item>ModDate</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Last Modified Date</p:item></p:name>
   <p:datatype><p:item>STRING(10)</p:item></p:datatype>
   <p:data><p:item>2012/05/06</p:item></p:data>
</p:fileInfoMember>
```

(4) Item description of ModTime element

Table 5-15 Item Description Specifications of ModTime Element

No.	Item Name	Description Details
1.	LABEL	Write "ModTime"
2.	LABEL2	Write "ModTime" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Last Modified Time" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(8)"
6.	DATA	Write the last modification time. Write the time in the format hh:mm:ss. Do not suppress any zeros in the hour, minute, or second.

(a) Item description example of ModTime element

```
<p:fileInfoMember label="ModTime">
   <p:label2><p:item>ModTime</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Last Modified Time</p:item></p:name>
   <p:datatype><p:item>STRING(8)</p:item></p:datatype>
   <p:data><p:item>11:22:33</p:item></p:data>
</p:fileInfoMember>
```

(5) Item description of Language element

Table 5-16 Item Description Specifications of Language Element

No.	Item Name	Description Details
1.	LABEL	Write "Language"
2.	LABEL2	Write "Language" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Supported Language" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(12)"
6.	DATA	Write the information of the language in which CSP+ is written. The following shows a list of defined language information as of May 2012: ・ Japanese: "ja" ・ English: "en" ・ Chinese (simplified Chinese): "zh-Hans" ・ Korean: "ko" ・ Chinese (unsimplified Chinese): "zh-Hant" ・ German: "de" ・ French: "fr" ・ Russian: "ru" ・ Italian: "it" ・ Spanish: "es" ・ Portuguese: "pt" When you want to use a language code other than the above, you need to submit an application to the CC-Link Partner Association. For an updated list of language information, contact the CC-Link Partner Association.

(a) Item description example of the Language element

```
<p:fileInfoMember label="Language">
   <p:label2><p:item>Language</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Supported Language</p:item></p:name>
   <p:datatype><p:item>STRING(12)</p:item></p:datatype>
   <p:data><p:item>ja</p:item></p:data>
</p:fileInfoMember>
```

(6) Item description of CCLinkFamilyProfileVersion element

Usable conventions varies depending on CSP+ Specifications Version described in this element. See Chapter 8 for details.

Table 5-17 Item Description Specifications of CCLinkFamilyProfileVersionElement

No.	Item Name	Description Details
1.	LABEL	Write "CCLinkFamilyProfileVersion"
2.	LABEL2	Write "CCLinkFamilyProfileVersion" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "CSP+ Specification Version" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(32)"
6.	DATA	Write the CSP+ specification version. For the CSP+ specification version, see Chapter 1. Note that the CSP+ specification version is expressed using two (alpha)numeric characters separated by a dot ("."), such as "x.y". x: Indicates the major version. y: Indicates the minor version. We recommend that the utility software execute operations in accordance with defined specifications, even if the minor version is wrong. The minor version may be omitted if the minor version is "0".

(a) Item description example of CCLinkFamilyProfileVersion element

```
<p:fileInfoMember label="CCLinkFamilyProfileVersion">
   <p:label2><p:item>CCLinkFamilyProfileVersion</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>CSP+ Specification version</p:item></p:name>
   <p:datatype><p:item>STRING(32)</p:item></p:datatype>
   <p:data><p:item>3.0</p:item></p:data>
</p:fileInfoMember>
```

(7) Item description of FileVersion element

Table 5-18 Item Description Specifications of FileVersionElement

No.	Item Name	Description Details
1.	LABEL	Write "FileVersion"
2.	LABEL2	Write "FileVersion" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "File Version" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(32)"
6.	DATA	Write the file version. Write the version using two (alpha)numeric characters separated by a dot ("."), such as "x.y". x: Describes the major version. A higher major version for the same module indicates that the file is newer. If the module is the same and the description information is the same for CSP+ files having different supported language information, the value of the major number is used to determine whether or not the CSP+ files are the same. y: Describes the minor version. A higher minor version with the same modules, supported languages, and major versions indicates that the file is newer. The minor version may be omitted if the minor version is "0". If the major version is increased, the minor version returns to 0. Table 5-19 indicates the file version upgrade rules.

Table 5-19 File Version Upgrade Rules

No.	Rule	Remarks
1.	Write a different file version for a CSP+ file when you change the contents of the file but the file has the same device version and supported language information.	
2.	If the CSP+ files have the device versions, write a higher value for the file version of the newer file.	
3.	If you upgrade the device version, set the major version of the file version to 1.	
4.	Write the same major number for the file versions of CSP+ files having the same contents but different supported language information.	
5.	Upgrade the minor version if you upgrade only language dependent information and do not need to upgrade the CSP+ files of other supported language information. For any other modifications (including addition of elements, etc.), major version must be updated. (However, this excludes cases in which device version is updated.) • Modification of CATEGORY item, NAME item, COMMENT item • Modification of DATA item of FILE_INFO part	

(a) Item description example of FileVersion element

```
<p:fileInfoMember label="FileVersion">
   <p:label2><p:item>FileVersion</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>File Version</p:item></p:name>
   <p:datatype><p:item>STRING(32)</p:item></p:datatype>
   <p:data><p:item>1.0</p:item></p:data>
</p:fileInfoMember>
```

5.2 DEVICE Section

The DEVICE section comprises one DEVICE_INFO part only. The one part is written as a single XML element. A comment may also be written (option).

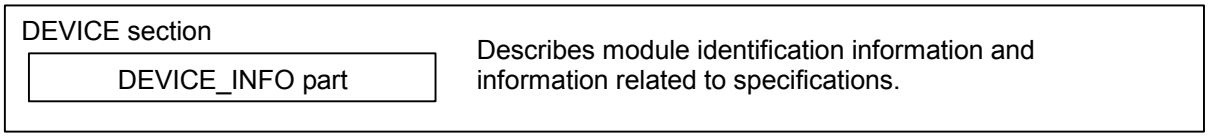


Figure 5-4 Structure of DEVICE Section

In XML format, the DEVICE section is written as an XML element of the XML element name "device". Table 5-20 shows a list of the attributes of the device element, and Table 5-21 shows a list of XML child elements.

Table 5-20 List of Attributes of device Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label that identifies the DEVICE section. (See Section 4.3.1.19.)	Required

Table 5-21 List of XML Child Elements of device Element

No.	XML Element Name	Description Details	Required/Optional
1.	label	Describes the comment related to the DEVICE section. (See Section 4.3.1.6.)	Optional
2.	deviceInfo	Describes the contents of the DEVICE_INFO part.	Required

5.2.1 DEVICE_INFO part

5.2.1.1 Overview

A DEVICE_INFO part describes product identification information and information related to product specifications.
Figure 5-5 shows the elements that make up the DEVICE_INFO part.
Figure 5-6 shows the elements that make up the DEVICE_INFO part "for tool link" in CC-Link IE TSN.

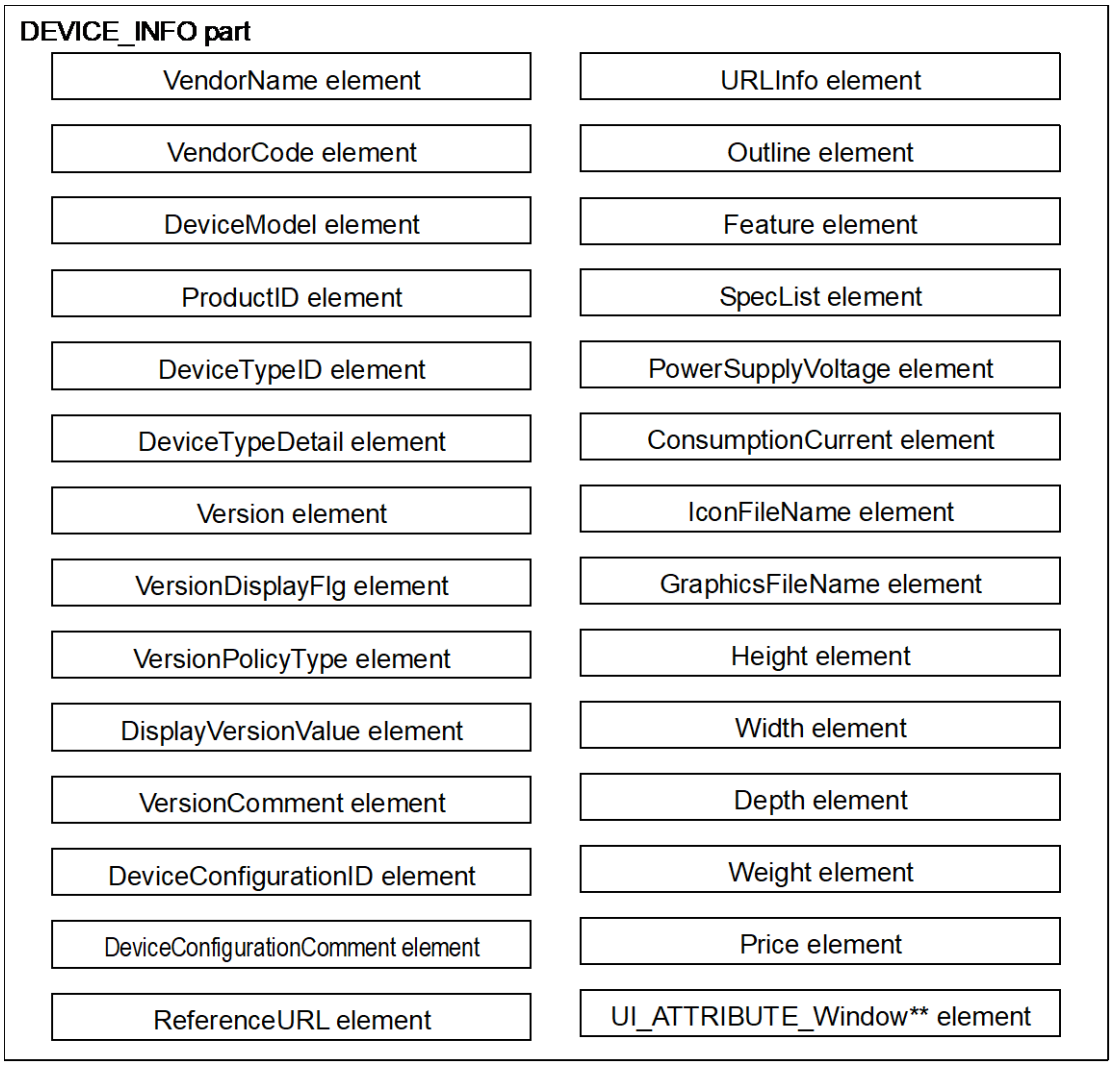


Figure 5-5 Structure of DEVICE_INFO Part

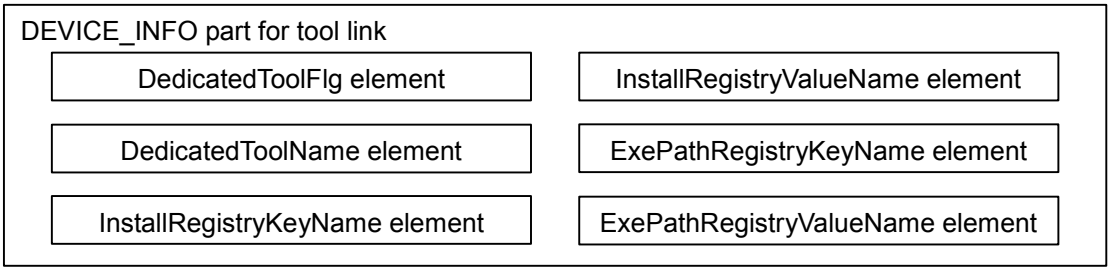


Figure 5-6 Structure of DEVICE_INFO Part for tool link in CC-Link IE TSN

The configuration of each element of the DEVICE_INFO part, that is, the items that should be described within each element, is the same. Figure 5-7 shows the configuration of an element of the DEVICE_INFO part.

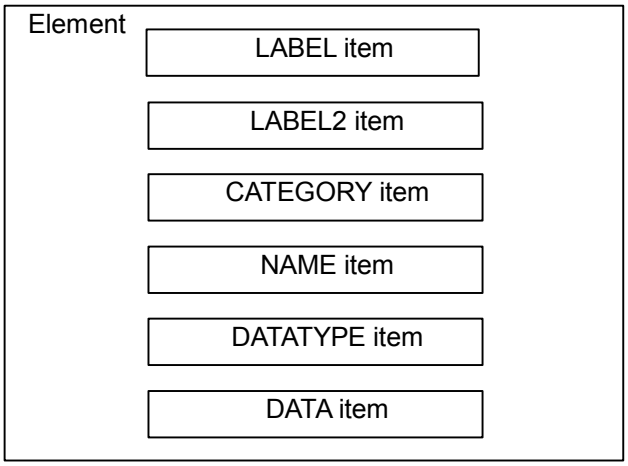


Figure 5-7 Structure of Element of DEVICE_INFO Part

Table 5-22 lists the elements. Table 5-23 lists the elements for tool link when the communication interface is CC-Link IE TSN. A single element is described as a single XML element. For details of the description specifications of each element, see Section 5.2.1.3.

Table 5-22 Element List for DEVICE_INFO Part

No.	Element	Description Details	Required/Optional
1.	VendorName	Describes the name of the vendor that manufactured the module.	Required
2.	VendorCode	Describes the vendor code of the vendor that manufactured the module. Describes the fifth to eighth digits of the partner member number of the CC-Link Partner Association.	Required
3.	DeviceModel	Describes the model of the module.	Required
4.	ProductID	Describes the product ID of the module. Describes the ID managed by the vendor that manufactured the module.	*2
5.	DeviceTypeID	Describes the ID that indicates the type of the module.	*1
6.	DeviceTypeDetail	Describes the specific device type.	*1
7.	Version	Describes the device version of the module.	Required
8.	VersionDisplayFlg	Describes whether the device version is to be shown to or hidden from the user.	Required
9.	VersionPolicyType	Describes the policy of the relationship between the mass production device version and the device version written in the CSP+ file when accessing the mass production device using the CSP+ file.	Required
10.	DisplayVersionValue	Describes the version value to be displayed when the value (Version) of the device version acquired from the mass production device differs from the version value displayed to the user on utility software.	Optional
11.	VersionComment	Describes the comment related to the device version.	Optional
12.	DeviceConfigurationID	Describes the device configuration ID.	Optional
13.	DeviceConfigurationComment	Describes the device configuration comment.	Optional
14.	ReferenceURL	Describes the URL when module information is disclosed on the Web.	Optional
15.	URLInfo	Describes an explanation of the information indicated by ReferenceURL.	Optional
16.	Outline	Describes the general specifications of the device.	Optional
17.	Feature	Describes the features of the device.	Optional
18.	SpecList	Describes the device specifications using a set of strings.	Optional
19.	PowerSupplyVoltage	Describes the power supply voltage in volts (V).	Optional
20.	ConsumptionCurrent	Describes the current consumption in milliamperes (mA).	Optional
21.	IconFileName	Describes the icon file name used when displaying the module as an icon with utility software, including the extension (.ico).	Optional
22.	GraphicsFileName	Describes the graphics file name used when the device is displayed with utility software, including the extension (.bmp, png, .jpg, .gif).	Optional
23.	Height	Describes the height of the external dimensions along with the unit of measurement.	Optional
24.	Width	Describes the width of the external dimensions along with the unit of measurement.	Optional
25.	Depth	Describes the depth of the external dimensions along with the unit of measurement.	Optional
26.	Weight	Describes the weight along with the unit of measurement.	Optional
27.	Price	Describes the price along with the unit of measurement.	Optional
28.	UI_ATTRIBUTE_Window**	Describes the name of the window specified by UI_ATTRIBUTE. Describes the window number in "**".	Optional

*1: Either the DeviceTypeID element or DeviceTypeDetail element must be written. Omitting both is prohibited. It is acceptable to write both, but in that case, the value of DeviceTypeDetail element takes precedence.

*2: This element is required in CC-Link IE TSN. This element is optional in other networks.

Table 5-23 Additional Element of DEVICE_INFO Part in CC-Link IE TSN

No.	Element	Description Details	Required/ Optional
1.	DedicatedToolFlg	Write whether or not there is a supported dedicated tool.	Optional
2.	DedicatedToolName	Write the tool name of the supported dedicated tool.	*1
3.	InstallRegistryKeyName	Write the existing registry key name only when the supported dedicated tool is installed.	*1
4.	InstallRegistryValueName	Write the existing registry value name only when the supported dedicated tool is installed.	*2
5.	ExePathRegistryKeyName	Write the registry key name that has the path information of the execution file (.exe) of the supported dedicated tool.	*1
6.	ExePathRegistryValueName	Write the value name of the registry where the path of the execution file (.exe) of the supported dedicated tool is written.	*1

*1: Required when DedicatedToolFlg is written and its DATA is 1. Otherwise, writing is prohibited.

*2: Optional when DedicatedToolFlg is written and its DATA is 1. Otherwise, writing is prohibited.

Table 5-24 shows the items described in the elements of the DEVICE_INFO part.

Table 5-24 Item List for DEVICE_INFO Part

No.	Item Name	Description Details	Description Specifications	Required /Optional
1.	LABEL	Describes the label for identifying the element.	See Section 4.3.1.19	Required
2.	LABEL2	Describes the label for identifying the element. (Used when the utility software supports another language.)	See Section 4.3.1.20	Optional
3.	CATEGORY	Describes the category for grouping the element.	See Section 4.3.1.4	Optional
4.	NAME	Describes the name of the element. Used when displaying the name and contents of the element in utility software.	See Section 4.3.1.24	Optional
5.	DATATYPE	Describes the data type of the contents described in the DATA item.	See Section 4.3.1.10	Required
6.	DATA	Describes the contents of the element.	See Section 4.3.1.9	Required

(1) DeviceTypeID element

If the device type does not fall under any of the categories, submit an application to the CC-Link Partner Associate to add a code.
Please contact the CC-Link Association.

(2) Device version (Version element)

The device version indicates the version of the function of the module. This information is used to associate the module and CSP+ file.

[Background]

When a module software version upgrade or the like is performed, specification changes, such as parameter or command additions, may occur. When module specifications are changed, the CSP+ file also needs to be updated in accordance with that change.

[Purpose of device version]

The information that identifies the state before and after a specification change is the device version. This information is used to indicate the correspondence between each CSP+ file and module specifications.

(a) Purpose of use by utility software

By managing and using with distinction all CSP+ files of different device versions, the utility software is capable of providing optimum functionality and UI in accordance with the version of the module used.

(b) Purpose of use by user

The user can select the CSP+ file for the device actually used by cross-checking the device version stated in the CSP+ file and the version of the module used.

[Data type]

The data type of the Version element is selectable per module from the following: bit string, signed integer, unsigned integer, or string. Be sure to use the same data type for the CSP+ files of different version corresponding to modules of the same model name.

[Description format]

The description format of the device version can be freely determined for each module. Since this value is used as an identifier in utility software and for automatic identification of the mass production version, the values of the items related to the device version need to be managed and written in accordance with the policies below.

(a) Updating the value in association with the module version

The device version indicates the version of the module with which the CSP+ file is used in combination. For this reason, the device version described in the CSP+ file and the module version need to be

managed in association with each other. Note, however, that the CSP+ file can be managed without upgrading the device version even if the module is one in which the device version is upgraded. [See (h).]

* The version of the module software/firmware may be written as the device version. However, when a defect is repaired, etc., these versions are upgraded even if the module function does not change (even if there is no need to update the CSP+ file).
  In the case of the example in Figure 5-8, when the device version is managed using the software version, the device version described in the CSP+ file needs to be upgraded when a defect is corrected, even if there is no functional change. Thus, if possible, we recommend that you do not use the software version for the device version, but rather a string that indicates the version of the function of the module.

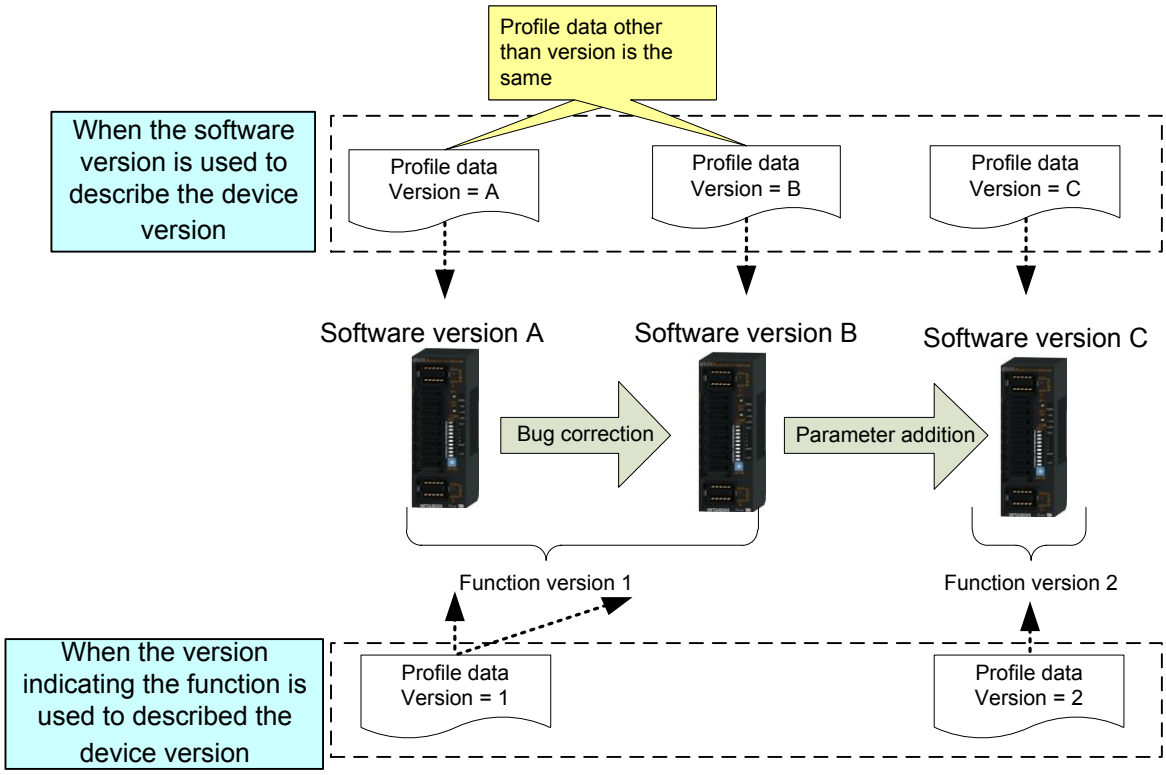


Figure 5-8 Device Version Management Example

(b) Relationship with version value held by module

It is possible to specify a method that acquires the version of the module using the utility software so that the version is indicated in the ReadVersionType element described in the COMM-IF_INFO part (see Section 5.3.1) described later. When the module is capable of acquiring the version value of the module, the value that can be actually acquired from the module is written in the DATA item of the VERSION element of the DEVICE_INFO part, and the data type of that value is written in the DATATYPE item. With this arrangement, the version value of the module and the version value described in the CSP+ file can be compared by the utility software. If the ReadVersionType element is set to "NA", there is no need to match the value with the version value held by the module.

(c) When the module is never subjected to version upgrades

Even if the module is not subjected to version upgrades and the concept of version does not exist in relation thereto, a value must be written in the Version element. (Write a value such as "1" or "A".)

(d) Version value displayed to user of module (DisplayVersion element, VersionComment element)

As shown in (b), in a case where the module and utility software are connected and the version value can be acquired, the version value is automatically assessed. However, even with such a module, if the module is not connected, the user must select and use the CSP+ file of the appropriate device version. In order to select the version, the version value needs to be displayed. The value of the Version element is written as the value held by the module as is, and thus the user may not be able to determine the version even upon viewing that value. When the version value held by the module differs from the value that should be displayed to the user in this case, a string for display can be written in the DisplayVersion element.

If the meaning of the version is unclear by the displayed version value alone, it is recommended that you describe the meaning of the version of the module using the VersionComment element.

(e) Showing/Hiding the device version value (VersionDisplaFlg element)

You can set whether or not the device version value needs to be displayed with the utility software using the VersionDisplayFlg element. 0 indicates hide, and 1 indicates show.
With a module that is not subject to version management or a module that does not disclose the device version value to the user, there is no needs to display the device version value using the utility software. Set the setting to hide.
When the values are mixed for a CSP+ of the same model names (when the show/hide specification changes midstream), the show or hide specification of higher priority is prioritized. The priority order is 1 > 0. While it is not prohibited to change the value when CSP+ is updated when the device version increases, etc., it is recommended since the value of lower priority is ignored as described above.

(f) Device version comparison policy for module and CSP+ file (VersionPolicyType element)

The VersionPolicyType element is used to indicate the policy of device versions with respect to the module and CSP+ file when the module is to be accessed using a CSP+ file. In a case where the utility software can automatically assess the device version of the module and the version written in the CSP+ file, errors and alarms are assessed in accordance with the value of the VersionPolicyType element. Even if the module is one that is not capable of automatic assessment, the user of the module needs to assess the device version that should be used based on this value. It is therefore recommended that the utility software indicates the meaning of this value (the value, in particular, needs to be a value other than 0).

Table 5-25 Meaning of Each Value of VersionPolicy and Example of Modules that Use the Value

Value	Meaning	Example of Module
0	The device version always uses the latest CSP+ file, regardless of the module device version	• When basically no problems will occur even if the device versions differ between the module and CSP+ file • When there is no need for the user to be aware of the device version (no need for awareness)
1	The CSP+ file having the exact same device version as the module is used	• When problems may potentially occur if the device versions differ between the module and CSP+ file • When the CSP+ files corresponding to all device versions are always disclosed
2	The CSP+ file having the latest device version, which is earlier than the module version, is used	• When all old device version functions are to be kept with the new device version (when treating the module as an old module is unproblematic) • When device versions are upgraded frequently due to bug repair, etc., and it is difficult to always disclose the CSP+ files that correspond to all device versions

* When VersionDisplayFlg is 0 (hide), the VersionPolicyTypemust be 0 as well.

* When the above-described values are mixed in the CSP+ files of the same model (when a policy changes midterm), the policy having the higher priority level is given higher priority. The priority level sequence is 1 > 2 > 0.

* While it is not prohibited to change the value when profile data is updated, it is not recommended since the value of lower priority is ignored as described above.

(g) Device version value comparison specifications

The newer/older device version comparison conducted according to VersionPolicy of (f) is implemented by the method described below.

- When the data type of Version is integer or bit string
  Performed by numerical value comparison. (Example: 1 < 9 < 10)
- When the data type of Version is string
  Performed by lexicographic comparison. (Example: 1 < 10 < 9)

(h) Management of CSP+ file in which the device version is not upgraded

Even with a module in which the software version is upgraded, it is possible to manage the CSP+ file without upgrading the device version. In such a case, CSP+ file version management is performed only with the file version [see Section 5.1.1.3(7)].

The advantages and disadvantages to not upgrading the device version are described below. For modules that will be highly problematic as a result of the disadvantages, we recommend that you do upgrade and manage the device version.

[Advantages to not upgrading the device version]

- The CSP+ file creator and manager do not need to manage the profiles of multiple device versions (just simply manage one file).
- The user does not need to be shown the device version (the user does not need to be made aware of version differences).

[Disadvantages to not upgrading the device version]

- The risk exists that inapplicable functions and parameters will be displayed by the utility software.
  Utility software uses the CSP+ file of the latest file version, regardless of the version of the module. As a result, even if an older module is used, the newer CSP+ file will be used, causing inapplicable functions and parameters to be displayed on the older module.
  If the user uses such functions and parameters without realizing it, problems will arise, such as error occurrence on the module or the execution of an operation not intended by the user.
  For modules that do not guarantee old/new compatibility, we recommended that do upgrade the device version.

5.2.1.2 XML format of DEVICE_INFO part

In XML format, the DEVICE_INFO part is described as an XML element having the XML element name "deviceInfo". Table 5-26 shows a list of the attributes of the deviceInfo element, and Table 5-27 shows a list of the XML child elements.

Table 5-26 Attribute List for deviceInfo Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the DEVICE_INFO part. (See Section 4.3.1.19.) (Example: "DeviceInformation")	Required

Table 5-27 XML Child Element List for deviceInfo Element

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the DEVICE_INFO part (See Section 4.3.1.6.)	Optional
2.	deviceInfoMember	Describes the contents of the elements of the DEVICE_INFO part	Required

In XML format, the element of the DEVICE_INFO part is described as an XML element having the XML element name "deviceInfoMember".

Table 5-28 shows a list of attributes of the deviceInfoMember element, and Table 5-29 shows a list of XML child elements.

Table 5-28 Attribute List for deviceInfoMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element.(See Section 4.3.1.19.) The contents written for this label are regulated by CSP+ specifications. For details, see Section 5.2.1.3.	Required

Table 5-29 XML Child Element List of deviceInfoMember Element

No.	XML Element Name	Description Details
1.	label2	Describes the LABEL2 item
2.	category	Describes the CATEGORY item
3.	name	Describes the NAME item
4.	datatype	Describes the DATATYPE item
5.	data	Describes the DATA item

Write the DEVICE_INFO part using a format such as shown below.

```
<p:deviceInfo label=''Write in accordance with the description details of Table 5-26
'Attribute Name label'''>
   <p:comment>''Write in accordance with the description details of Table 5-27 'XML Element
   Name comment'''</p:comment>
   <p:deviceInfoMember label=''Write in accordance with the description details of Table
   5-28 'Attribute Name label'''>
      <p:label2>''Write in accordance with the description details of Table 5-29 'XML Element
      Name label2'''</p:label2>
      <p:category>''Write in accordance with the description details of Table 5-29 'XML
      Element Name category'''</p:category>
      <p:name>''Write in accordance with the description details of Table 5-29 'XML Element
      Name name''</p:name>
      <p:datatype>''Write in accordance with the description details of Table 5-29 'XML
      Element Name datatype''</p:datatype>
      <p:data>''Write in accordance with the description details of Table 5-29 'XML Element
      Name data'''</p:data>
   </p:deviceInfoMember>
   <p:deviceInfoMember label=''Write in accordance with the description details of Table
   5-28 'Attribute Name label'''>
   :
   :
   </p:deviceInfoMember>
   :
   :
</p:deviceInfo>
```

5.2.1.3 Description details of DEVICE_INFO part

(1) Item description of VendorName element

Table 5-30 Item Description Specifications of VendorName Element

No.	Item Name	Description Details
1.	LABEL	Write "VendorName"
2.	LABEL2	Write "VendorName" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Vendor Name" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(64)"
6.	DATA	Write the name of the vendor that manufactured the module

(a) Item description example of VendorName element

```
<p:deviceInfoMember label="VendorName">
   <p:label2><p:item>VendorName</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Vendor Name</p:item></p:name>
   <p:datatype><p:item>STRING(64)</p:item></p:datatype>
   <p:data><p:item>CLPA</p:item></p:data>
</p:deviceInfoMember>
```

(2) Item description of VendorCode element

Table 5-31 Item Description Specifications of VendorCode Element

No.	Item Name	Description Details
1.	LABEL	Write "VendorCode"
2.	LABEL2	Write "VendorCode" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Vendor Code" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "WORD"
6.	DATA	Write the vendor code The fifth to eighth digits of the partner member number of the CC-Link Partner Association is the vendor code. In the case of the CC-Link IE TSN support unit which has controller information, write the vendor code of the controller information. In the case of the device which does not have controller information, write the vendor code of the network information.

(a) Item description example of VendorCode element

```
<p:deviceInfoMember label="VendorCode">
   <p:label2><p:item>VendorCode</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Vendor Code</p:item></p:name>
   <p:datatype><p:item>WORD</p:item></p:datatype>
   <p:data><p:item>0x0000</p:item></p:data>
</p:deviceInfoMember>
```

(3) Item description of DeviceModel element

DeviceModel is information for specifying a module. If you change the value of this element at the time of CSP+ updating, the CSP+ is treated as a CSP+ of a different module.

Table 5-32 Item Description Specifications of DeviceModel Element

No.	Item Name	Description Details
1.	LABEL	Write "DeviceModel"
2.	LABEL2	Write "DeviceModel" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Device Model" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(48)"
6.	DATA	Write the model of the module managed by each vendor. The model is used in the file name (see Section 4.1.1) of the CSP+ file, and thus the characters that are prohibited in the file name [¥/:?"<>|] cannot be used. If model assessment processing is performed by the utility software, there is no upper/lower case sensitivity.

(a) Item description example of the DeviceModel element

```
<p:deviceInfoMember label="DeviceModel">
   <p:label2><p:item>DeviceModel</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Device Model</p:item></p:name>
   <p:datatype><p:item>STRING(48)</p:item></p:datatype>
   <p:data><p:item>CC-A1000</p:item></p:data>
</p:deviceInfoMember>
```

(4) Item description of ProductID element

When you write the Product ID, write a unique ID having a one-to-one correspondence with the module model name for each vendor. Write the same ProductID for all CSP+ files of different versions of the same model names; do not change the ID.

Table 5-33 Item Description Specifications of ProductID Element

No.	Item Name	Description Details
1.	LABEL	Write "ProductID"
2.	LABEL2	Write "ProductID" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write the "Product ID" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(256)"
6.	DATA	Write the product ID of the module managed by the vendor. In the case of the CC-Link IE Field Network Safety Communication Function Support Unit, enter the value of unit_type_code. For unit_type_code, see "CC-Link IE Safety Communication Function Specification (Application Layer Service and Protocol) (Communication Profile) BAP-C2007ENG-002, BAP-C2007ENG-003". When it is possible to acquire the type name code of the controller in the CC-Link IE Field Network Support Unit, be certain to write the same value as that value. For the type name code of the controller, see "CC-Link IE Field Network Specification (Application Layer Protocol) BAP-C2005-004". In the case of the Ethernet connection unit, write the type name code with a string of hexadecimal representation granted "0x" at the head. For the type name code, see "SLMP (Seamless Message Protocol) Specification (Protocol) BAP-C2006ENG-003" and "CC-Link IE Field Network Basic Specification (Application Layer Protocol) BAP-C2010ENG-004". In the case of the CC-Link IE TSN support unit which has controller information, write the type name code of the controller information in decimal. In the case of the device which does not have controller information, write the type name code of the network information in decimal. This item is required in CC-Link IE TSN.

(a) Item description example of ProductID element

```
<p:deviceInfoMember label="ProductID">
   <p:label2><p:item>ProductID</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Product ID</p:item></p:name>
   <p:datatype><p:item>STRING(256)</p:item></p:datatype>
   <p:data><p:item>0x00000001</p:item></p:data>
</p:deviceInfoMember>
```

(5) Item description of DeviceTypeID element

Table 5-34 Item Description Specifications of DeviceTypeID Element

No.	Item Name	Description Details
1.	LABEL	Write "DeviceTypeID".
2.	LABEL2	Write the "DeviceTypeID" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write the "Device Type ID" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "WORD"
6.	DATA	Write the ID that indicates the type of the module. For IDs, see Section 5.2.1.1(1).

(a) Item description example of the DeviceTypeID element

```
<p:deviceInfoMember label=DeviceTypeID>
   <p:label2><p:item>DeviceTypeID</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Device Type ID</p:item></p:name>
   <p:datatype><p:item>WORD</p:item></p:datatype>
   <p:data><p:item>0x04</p:item></p:data>
</p:deviceInfoMember>
```

(6) Item description of DeviceTypeDetail element

Table 5-35 Item Description Specifications of DeviceTypeDetail Element

No.	Item Name	Description Details
1.	LABEL	Write "DeviceTypeDetail"
2.	LABEL2	Write "DeviceTypeDetail" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Device Type Detail" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING_U(256)"
6.	DATA	Write the details of the device type. Write this value when you want to indicate a specific module type that cannot be expressed using DeviceTypeID. We recommend that you write a string that indicates the module type using this element alone.

(a) Item description example of DeviceTypeDetail element

```
<p:deviceInfoMember label="DeviceTypeDetail">
   <p:label2><p:item>DeviceTypeDetail</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Device Type Detail</p:item></p:name>
   <p:datatype><p:item>STRING(256)</p:item></p:datatype>
   <p:data><p:item>Inverter A1234</p:item></p:data>
</p:deviceInfoMember>
```

(7) Item description of Version element

Table 5-36 Item Description Specifications of Version Element

No.	Item Name	Description Details
1.	LABEL	Write "Version"
2.	LABEL2	Write "Version" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Device Version" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write the value selecting from the following: bit string type, signed integer type, unsigned integer type, and string type [using only the expression "STRING(x)"]. In the following example, "STRING(16)" is written. In the case of Ethernet connection unit: Write "WORD". In the case of the CC-Link IE TSN connection unit: Write "UINT16".
6.	DATA	Write the function version of the module The module function version is used to associate each version and profile of the module when profile description changes are required, such as parameter and command changes. The description format may be freely determined for each module. However, since the device version is quoted in the CSP+ file name (see Section 4.1.1), do not use characters that are prohibited in the file name [¥/:?"<>|]. For details, see Section 5.2.1.1(2). In the case of the CC-Link IE Field Network Safety Communication Function Support Unit, write the value of the unit_version. For unit_version, see "CC-Link IE Safety Communication Function Specification (Application Layer Service and Protocol) (Communication profile) BAP-C2007ENG-002, BAP-C2007ENG-003". In the case of the Ethernet connection unit, write a value equivalent to the serverMachineVersion item stored in the answer data ResNodeSearch of "SLMP (Seamless Message Protocol) Specification (Protocol) BAP-C2006ENG-003". In the case of the CC-Link IE TSN support unit which has controller information, write the device version of the controller information. In the case of the device which does not have controller information, write the device version of the network information. For the device version of the controller information, see "CC-Link IE TSN Specifications (Overview) BAP-C2011ENG-001".

(a) Item description example of Version element

```
<p:deviceInfoMember label="Version">
   <p:label2><p:item>Version</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Device Version</p:item></p:name>
   <p:datatype><p:item>STRING(16)</p:item></p:datatype>
   <p:data><p:item>1.00A</p:item></p:data>
</p:deviceInfoMember>
```

(8) Item description of VersionDisplayFlg element

Table 5-37 Item Description Specifications of VersionDisplayFlg Element

No.	Item Name	Description Details
1.	LABEL	Write "VersionDisplayFlg ".
2.	LABEL2	Write "VersionDisplayFlg " in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Device Version Display Flag" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether or not the device version is to be shown to or hidden from the user. Write the value based on the following [0: Hide, 1: Show]. For details, see Section 5.2.1.1(2) [Description Format] (e).

(a) Item description example of VersionDisplayFlg element

```
<p:deviceInfoMember label="VersionDisplayFlg">
   <p:label2><p:item>VersionDisplayFlg</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Device Version Display Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:deviceInfoMember>
```

(9) Item description of VersionPolicyType element

Table 5-38 Item Description Specifications of VersionPolicyType Element

No.	Item Name	Description Details
1.	LABEL	Write "VersionPolicyType"
2.	LABEL2	Write "VersionPolicyType" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Device Version Display Policy" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "UINT16"
6.	DATA	Write the policy regarding the relationship between the device version of the module and the device version written in the CSP+ file when the mass production device is to be accessed using the CSP+ file. [See Section 5.2.1.1(2).]

(a) Item description example of VersionPolicyType element

```
<p:deviceInfoMember label="VersionPolicyType">
   <p:label2><p:item>VersionPolicyType</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Device Version Display Policy</p:item></p:name>
   <p:datatype><p:item> UINT16</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:deviceInfoMember>
```

(10) Item description of DisplayVersionValue element

Table 5-39 Item Description Specifications of DisplayVersionValue Element

No.	Item Name	Description Details
1.	LABEL	Write "DisplayVersionValue"
2.	LABEL2	Write "DisplayVersionValue" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Device Version (Display Value)" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(32)"
6.	DATA	Write the value of the version to be displayed when the value of the device version acquired from the module differs from the value of the version to be displayed to the user with the utility software. [See Section 5.2.1.1(2).] This may be omitted if the value is the same as the value of the Version element.

(a) Item description example of DisplayVersionValue element

```
<p:deviceInfoMember label="DisplayVersionValue">
   <p:label2><p:item>DisplayVersionValue</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Device Version (Display Value)</p:item></p:name>
   <p:datatype><p:item>STRING(32)</p:item></p:datatype>
   <p:data><p:item>1.00A</p:item></p:data>
</p:deviceInfoMember>
```

(11) Item description of VersionComment element

Table 5-40 Item Description Specifications of VersionComment Element

No.	Item Name	Description Details
1.	LABEL	Write "VersionComment"
2.	LABEL2	Write "VersionComment" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Version Comment" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING_U(256)"
6.	DATA	Write the device version comment. [See Section 5.2.1.1(2).]

(a) Item description example of VersionComment element

```
<p:deviceInfoMember label="VersionComment">
   <p:label2><p:item>VersionComment</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Version Comment</p:item></p:name>
   <p:datatype><p:item>STRING_U(256)</p:item></p:datatype>
   <p:data><p:item>Customized for Company A</p:item></p:data>
</p:deviceInfoMember>
```

(12) Item description of DeviceConfigurationID element

Table 5-41 Item Description Specifications of DeviceConfigurationID Element

No.	Item Name	Description Details
1.	LABEL	Write "DeviceConfigurationID"
2.	LABEL2	Write "DeviceConfigurationID" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Device Configuration ID" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(32)"
6.	DATA	Write the identifier for the individual module configuration of the module having a device configuration that can be changed by the user. The prohibited characters are the same as those of the conventions for the LABEL item. (See Section 4.3.1.19.) A different CSP+ is identified based on the existence or nonexistence of a device configuration ID and the difference in the written string. Users who determine the device configuration ID need to manage the CSP+ files so that the ID is unique within the environment used by the CSP+ file. There is no need to standardize the management of file versions between profiles having different device configuration IDs. It is possible to create and operate multiple different profiles for one module by using the device configuration ID. This is possible by with modules such as described in Section (3).

(a) Item description example of DeviceConfigurationID element

```
<p:deviceInfoMember label="DeviceConfigurationID">
   <p:label2><p:item>DeviceConfigurationID</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Device Configuration ID</p:item></p:name>
   <p:datatype><p:item>STRING(32)</p:item></p:datatype>
   <p:data><p:item>A_Customize</p:item></p:data>
</p:deviceInfoMember>
```

(13) Item description DeviceConfigurationComment element

Table 5-42 Item Description Specifications of DeviceConfigurationComment Element

No.	Item Name	Description Details
1.	LABEL	Write "DeviceConfigurationComment"
2.	LABEL2	Write "DeviceConfigurationComment" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Device Configuration Comment" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING_U(64)"
6.	DATA	Write a string for display for user identification of the profile to be identified by the device configuration ID.

(a) Item description example of DeviceConfigurationComment element

```
<p:deviceInfoMember label="DeviceConfigurationComment">
   <p:label2><p:item>DeviceConfigurationComment</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Device Configuration Comment</p:item></p:name>
   <p:datatype><p:item>STRING_U(64)</p:item></p:datatype>
   <p:data><p:item>Customized for Company A</p:item></p:data>
</p:deviceInfoMember>
```

(14) Item description of ReferenceURL element

Table 5-43 Item Description Specifications of ReferenceURL Element

No.	Item Name	Description Details
1.	LABEL	Write "ReferenceURL".
2.	LABEL2	Write "ReferenceURL" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON".
4.	NAME	Write "Reference URL" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	There are no restrictions on the number of characters of the data type Write a value that is larger than the number of characters described for "data". In the following example, "STRING_U(1024)" is written.
6.	DATA	Write the URL when the module information is disclosed on the Web If the module information is not directly disclosed, write the Web page of the vendor.

(a) Item description example of ReferenceURL element

```
<p:deviceInfoMember label="ReferenceURL">
   <p:label2><p:item>ReferenceURL</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Reference URL</p:item></p:name>
   <p:datatype><p:item>STRING_U(1024)</p:item></p:datatype>
   <p:data>
      <p:item>http://wwwf2.mitsubishielectric.co.jp/melfansweb/index_j.htm</p:item>
   </p:data>
</p:deviceInfoMember>
```

(15) Item description of URLInfo element

Table 5-44 Item Description Specifications of URLInfo Element

No.	Item Name	Description Details
1.	LABEL	Write "URLInfo"
2.	LABEL2	Write "URLInfo" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "URL Info" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING_U(256)"
6.	DATA	Write a description of the information indicating ReferenceURL.

(a) Item description example of URLInfo element

```
<p:deviceInfoMember label="URLInfo">
   <p:label2><p:item>URLInfo</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>URL Info</p:item></p:name>
   <p:datatype><p:item>STRING_U(256)</p:item></p:datatype>
   <p:data><p:item>Check Detail on MELFANS web</p:item></p:data>
</p:deviceInfoMember>
```

(16) Item description of Outline element

Table 5-45 Item Description Specifications of Outline Element

No.	Item Name	Description Details
1.	LABEL	Write "Outline"
2.	LABEL2	Write "Outline" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Outline" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING_U(256)"
6.	DATA	Write the general specifications of the module.

(a) Item description example of Outline element

```
<p:deviceInfoMember label="Outline">
   <p:label2><p:item>Outline</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Outline</p:item></p:name>
   <p:datatype><p:item>STRING_U(256)</p:item></p:datatype>
   <p:data><p:item>A/D Converter 4CH 12-bit</p:item></p:data>
</p:deviceInfoMember>
```

(17) Item description of Feature element

Table 5-46 Item Description Specifications of Feature Element

No.	Item Name	Description Details
1.	LABEL	Write "Feature"
2.	LABEL2	Write "Feature" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Feature" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING_U(256)"
6.	DATA	Write the module features

(a) Item description example of Feature element

```
<p:deviceInfoMember label="Feature">
   <p:label2><p:item>Feature</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Feature</p:item></p:name>
   <p:datatype><p:item>STRING_U(256)</p:item></p:datatype>
   <p:data><p:item>2 Channel</p:item></p:data>
</p:deviceInfoMember>
```

(18) Item description of SpecList element

Table 5-47 Item Description Specifications of SpecList Element

No.	Item Name	Description Details
1.	LABEL	Write "SpecList"
2.	LABEL2	Write "SpecList" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Spec List" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING_U(256)()"
6.	DATA	Write the module specification information as a set

(a) Item description example of SpecList element

```
<p:deviceInfoMember label="SpecList">
   <p:label2><p:item>SpecList</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Spec List</p:item></p:name>
   <p:datatype><p:item>STRING_U(256)()</p:item></p:datatype>
   <p:data>
      <p:item>Total Accuracy ±1%</p:item>
      <p:item>Maximum conversion Speed: 1 ms/channel</p:item>
   </p:data>
</p:deviceInfoMember>
```

(19) Item description of PowerSupplyVoltage element

Table 5-48 Item Description Specifications of PowerSupplyVoltage Element

No.	Item Name	Description Details
1.	LABEL	Write "PowerSupplyVoltage"
2.	LABEL2	Write "PowerSupplyVoltage" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Power Supply Voltage(V)" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "REAL"
6.	DATA	Write the power supply voltage (V)

(a) Item description example of PowerSupplyVoltage element

```
<p:deviceInfoMember label="PowerSupplyVoltage">
   <p:label2><p:item>PowerSupplyVoltage</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Power Supply Voltage(V)</p:item></p:name>
   <p:datatype><p:item>REAL</p:item></p:datatype>
   <p:data><p:item>24</p:item></p:data>
</p:deviceInfoMember>
```

(20) Item description of ConsumptionCurrent element

Table 5-49 Item Description Specifications of ConsumptionCurrent Element

No.	Item Name	Description Details
1.	LABEL	Write "ConsumptionCurrent"
2.	LABEL2	Write "ConsumptionCurrent" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Consumption Current (mA)" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "REAL"
6.	DATA	Write the consumption current (mA)

(a) Item description example of ConsumptionCurrent element

```
<p:deviceInfoMember label="ConsumptionCurrent">
   <p:label2><p:item>ConsumptionCurrent</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Consumption Current(mA)</p:item></p:name>
   <p:datatype><p:item>REAL</p:item></p:datatype>
   <p:data><p:item>120</p:item></p:data>
</p:deviceInfoMember>
```

(21) Item description IconFileName element

Table 5-50 Item Description Specifications of IconFileName Element

No.	Item Name	Description Details
1.	LABEL	Write "IconFileName"
2.	LABEL2	Write "IconFileName" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Icon File Name" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(52)"
6.	DATA	Write the icon file name to be used when a module icon is to be displayed using the utility software. The characters permitted in the file name are single-byte alphanumeric characters and symbols only. Include a 16x16 pixel icon in the ico file.

(a) Item description example of IconFileName element

```
<p:deviceInfoMember label="IconFileName">
   <p:label2><p:item>IconFileName</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Icon File Name</p:item></p:name>
   <p:datatype><p:item>STRING(52)</p:item></p:datatype>
   <p:data><p:item>AJ65BT-64AD.ico</p:item></p:data>
</p:deviceInfoMember>
```

(22) Item description of GraphicsFileName element

Table 5-51 Item Description Specifications of GraphicsFileName Element

No.	Item Name	Description Details
1.	LABEL	Write "GraphicsFileName"
2.	LABEL2	Write "GraphicsFileName" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Graphics File Name" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(52)"
6.	DATA	Write the graphics file name to be used when a module is to be displayed using the utility software. The characters permitted in the file name are single-byte alphanumeric characters and symbols only. The applicable file types are as follows: PNG file: .png BMP file: .bmp JPEG file: .jpg GIF file: .gif Create a bitmap size within the range of 32x32 to 256x256 pixels.

(a) Item description example of GraphicsFileName element

```
<p:deviceInfoMember label="GraphicsFileName">
   <p:label2><p:item>GraphicsFileName</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Graphics File Name</p:item></p:name>
   <p:datatype><p:item>STRING(52)</p:item></p:datatype>
   <p:data><p:item>AJ65BT-64AD.bmp</p:item></p:data>
</p:deviceInfoMember>
```

(23) Item description of Height element

Table 5-52 Item Description Specifications of Height Element

No.	Item Name	Description Details
1.	LABEL	Write "Height"
2.	LABEL2	Write "Height" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Height" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING_U(256)"
6.	DATA	Write the height of the external dimensions. Write the value along with the unit of measurement.

(a) Item description example of Height element

```
<p:deviceInfoMember label="Height">
   <p:label2><p:item>Height</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Height</p:item></p:name>
   <p:datatype><p:item>STRING_U(256)</p:item></p:datatype>
   <p:data><p:item>65mm</p:item></p:data>
</p:deviceInfoMember>
```

(24) Item description of Width element

Table 5-53 Item Description Specifications of Width Element

No.	Item Name	Description Details
1.	LABEL	Write "Width"
2.	LABEL2	Write "Width" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Width" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING_U(256)"
6.	DATA	Write the width of the external dimensions. Write the value along with the unit of measurement.

(a) Item description example of Width element

```
<p:deviceInfoMember label="Width">
   <p:label2><p:item>Width</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Width</p:item></p:name>
   <p:datatype><p:item>STRING_U(256)</p:item></p:datatype>
   <p:data><p:item>151.9mm</p:item></p:data>
</p:deviceInfoMember>
```

(25) Item description of Depth element

Table 5-54 Item Description Specifications of Depth Element

No.	Item Name	Description Details
1.	LABEL	Write "Depth"
2.	LABEL2	Write "Depth" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Depth" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING_U(256)"
6.	DATA	Write the depth of the external dimensions. Write the value along with the unit of measurement.

(a) Item description example of Depth element

```
<p:deviceInfoMember label="Depth">
   <p:label2><p:item>Depth</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Depth</p:item></p:name>
   <p:datatype><p:item>STRING_U(256)</p:item></p:datatype>
   <p:data><p:item>63mm</p:item></p:data>
</p:deviceInfoMember>
```

(26) Item description of Weight element

Table 5-55 Item Description Specifications of Weight Element

No.	Item Name	Description Details
1.	LABEL	Write "Weight"
2.	LABEL2	Write "Weight" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Weight" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING_U(256)"
6.	DATA	Write the weight. Write the value along with the unit of measurement.

(a) Item description example of Weight element

```
<p:deviceInfoMember label="Weight">
   <p:label2><p:item>Weight</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Weight</p:item></p:name>
   <p:datatype><p:item>STRING_U(256)</p:item></p:datatype>
   <p:data><p:item>0.35kg</p:item></p:data>
</p:deviceInfoMember>
```

(27) Item description of Price element

Table 5-56 Item Description Specifications of Price Element

No.	Item Name	Description Details
1.	LABEL	Write "Price"
2.	LABEL2	Write "Price" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Price" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING_U(256)"
6.	DATA	Write the price Write the value along with the unit of measurement.

(a) Item description example of Price element

```
<p:deviceInfoMember label="Price">
   <p:label2><p:item>Price</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Price</p:item></p:name>
   <p:datatype><p:item>STRING_U(256)</p:item></p:datatype>
   <p:data><p:item>90000 yen</p:item></p:data>
</p:deviceInfoMember>
```

(28) Item description of UI_ATTRIBUTE_Window** element

Table 5-57 Item Description Specifications of UI_ATTRIBUTE_Window Element**

No.	Item Name	Description Details
1.	LABEL	Write "UI_ATTRIBUTE_Window**". Enter the window number in the area marked "**".
2.	LABEL2	Write "UI_ATTRIBUTE_Window**" in the case of English CSP+ files. [Recommended specifications] Write ** so that it is the same as the LABEL item.
3.	CATEGORY	Write "COMMON".
4.	NAME	Write "Display Info **" in the case of English CSP+ files. [Recommended specifications] Write ** so that it is the same as the LABEL item.
5.	DATATYPE	Write "STRING_U(256)".
6.	DATA	Write the name of the window specified in UI_ATTRIBUTE.

(a) Item description example of UI_ATTRIBUTE_Window1 element

```
<p:deviceInfoMember label="UI_ATTRIBUTE_Window1">
   <p:label2><p:item>UI_ATTRIBUTE_Window1</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Display Info 1</p:item></p:name>
   <p:datatype><p:item>STRING_U(256)</p:item></p:datatype>
   <p:data><p:item>Window1</p:item></p:data>
</p:deviceInfoMember>
```

- Description of DEVICE_INFO part for tool link when the communication interface is CC-Link IE TSN

(29) Item description of DedicatedToolFlg element

Table 5-58 Item Description Specifications of DedicatedToolFlg Element

No.	Item Name	Description Details
1.	LABEL	Write "DedicatedToolFlg".
2.	LABEL2	Write "DedicatedToolFlg" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON".
4.	NAME	Write "Dedicated Tool Available/Unavailable Flag" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether or not there is a supported dedicated tool. 0: None 1: Available If omitted, 0 is regarded to be set.

(a) Item description example of DedicatedToolFlg element

```
<p:deviceInfoMember label="DedicatedToolFlg">
   <p:label2><p:item>DedicatedToolFlg</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Dedicated Tool Available/Unavailable Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:deviceInfoMember>
```

(30) Item description of DedicatedToolName element

Table 5-59 Item Description Specifications of DedicatedToolName Element

No.	Item Name	Description Details
1.	LABEL	Write "DedicatedToolName".
2.	LABEL2	Write "DedicatedToolName" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON".
4.	NAME	Write "Dedicated Tool Name" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "STRING_U(256)".
6.	DATA	Write the tool name of the supported dedicated tool.

(a) Item description example of DedicatedToolName element

```
<p:deviceInfoMember label="DedicatedToolName">
   <p:label2><p:item>DedicatedToolName</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>DedicatedToolName</p:item></p:name>
   <p:datatype><p:item>STRING_U(256)</p:item></p:datatype>
   <p:data><p:item>Dedicated tool</p:item></p:data>
</p:deviceInfoMember>
```

(31) Item description of InstallRegistryKeyName element

Table 5-60 Item Description Specifications of InstallRegistryKeyName Element

No.	Item Name	Description Details
1.	LABEL	Write "InstallRegistryKeyName".
2.	LABEL2	Write "InstallRegistryKeyName" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON".
4.	NAME	Write "Install Registry Key Name" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "STRING(255)".
6.	DATA	Write the existing registry key name only when the supported dedicated tool is installed. For the key name that can be written, only one of "HKEY_LOCAL_MACHINE" can be used. Write the key name of "HKEY_LOCAL_MACHINE¥" or after.

(a) Item description example of InstallRegistryKeyName element

```
<p:deviceInfoMember label="InstallRegistryKeyName">
   <p:label2><p:item>InstallRegistryKeyName</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Install Registry Key Name/p:item></p:name>
   <p:datatype><p:item>STRING(255)</p:item></p:datatype>
   <p:data><p:item>SOFTWARE¥AAA</p:item></p:data>
</p:deviceInfoMember>
```

(32) Item description of InstallRegistryValueName element

Table 5-61 Item Description Specifications of InstallRegistryValueName Element

No.	Item Name	Description Details
1.	LABEL	Write "InstallRegistryValueName".
2.	LABEL2	Write "InstallRegistryValueName" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON".
4.	NAME	Write "Install Registry Value Name" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "STRING(255)".
6.	DATA	Write the existing registry value name only when the supported dedicated tool is installed. This does not need to be written when installation can be determined from only the key name written in InstallRegistryKeyName.

(a) Item description example of InstallRegistryValueName element

```
<p:deviceInfoMember label="InstallRegistryValueName">
   <p:label2><p:item>InstallRegistryValueName</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Install Registry Value Name</p:item></p:name>
   <p:datatype><p:item>STRING(255)</p:item></p:datatype>
   <p:data><p:item>AppMain</p:item></p:data>
</p:deviceInfoMember>
```

(33) Item description of ExePathRegistryKeyName element

Table 5-62 Item Description Specifications of ExePathRegistryKeyName Element

No.	Item Name	Description Details
1.	LABEL	Write "ExePathRegistryKeyName".
2.	LABEL2	Write "ExePathRegistryKeyName" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON".
4.	NAME	Write "Exe Path Registry Key Name" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "STRING(255)".
6.	DATA	Write the registry key name that has the path information of the execution file (.exe) of the supported dedicated tool. Write the key name of "HKEY_LOCAL_MACHINE¥" or after.

(a) Item description example of ExePathRegistryKeyName element

```
<p:deviceInfoMember label="ExePathRegistryKeyName">
   <p:label2><p:item>ExePathRegistryKeyName</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Exe Path Registry Key Name</p:item></p:name>
   <p:datatype><p:item>STRING(255)</p:item></p:datatype>
   <p:data><p:item>SOFTWARE¥AAA</p:item></p:data>
</p:deviceInfoMember>
```

(34) Item description of ExePathRegistryValueName element

Table 5-63 Item Description Specifications of ExePathRegistryValueName Element

No.	Item Name	Description Details
1.	LABEL	Write "ExePathRegistryValueName".
2.	LABEL2	Write "ExePathRegistryValueName" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON".
4.	NAME	Write "Exe Path Registry Value Name" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "STRING(255)".
6.	DATA	Write the value name of the registry where the path of the execution file (.exe) of the supported dedicated tool is written.

(a) Item description example of ExePathRegistryValueName element

```
<p:deviceInfoMember label="ExePathRegistryValueName">
   <p:label2><p:item>ExePathRegistryValueName</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Exe Path Registry Value Name</p:item></p:name>
   <p:datatype><p:item>STRING(255)</p:item></p:datatype>
   <p:data><p:item>AppMain</p:item></p:data>
</p:deviceInfoMember>
```

5.3 COMM_IF Section

The COMM_IF section comprises multiple parts as shown below. One part is written as a single XML element. A comment may also be written (option).

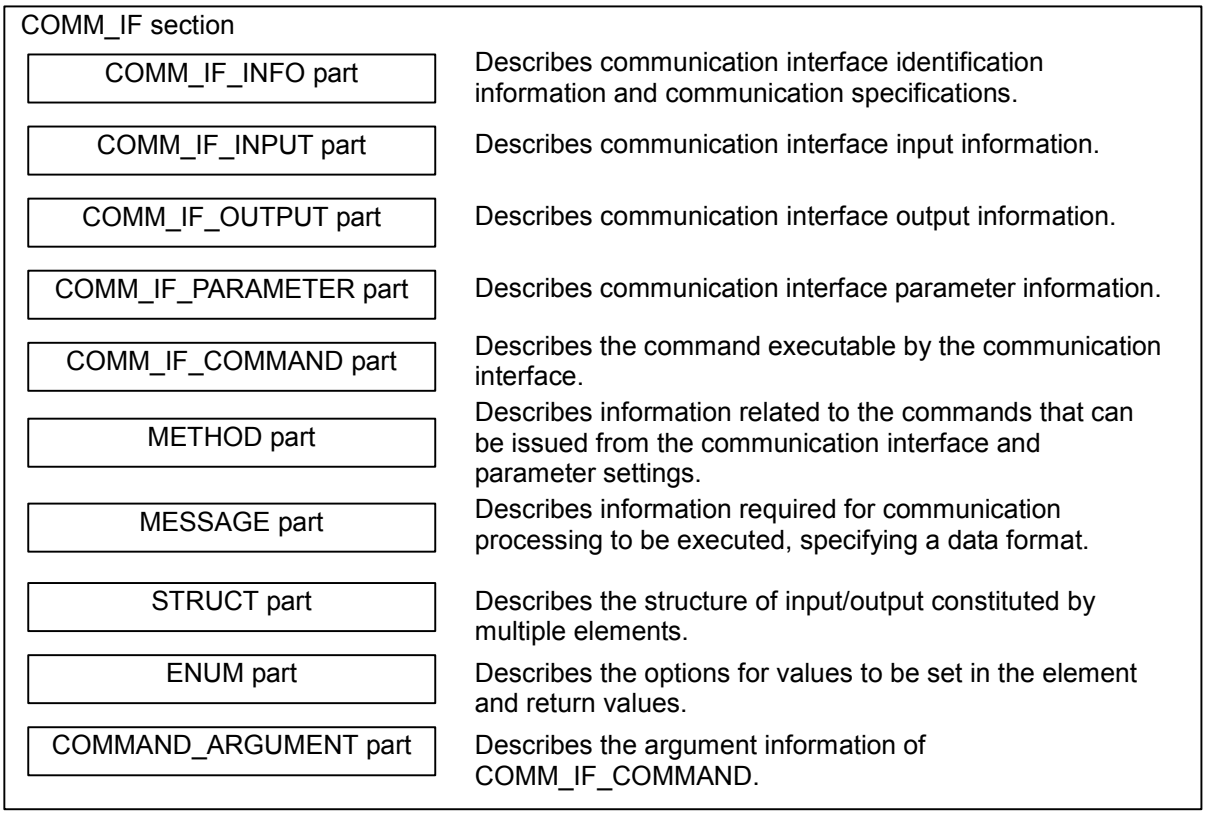


Figure 5-9 Structure of COMM_IF Section

In XML format, the COMM_IF section is written as an XML element of the XML element name "commIf". Table 5-64 shows a list of the attributes of the commIf element, and Table 5-65 shows a list of XML child elements.

Table 5-64 List of Attributes of commIf Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label that identifies the COMM_IF section. (See Section 4.3.1.19.) (Example: "CommIfSection")	Required

Table 5-65 List of XML Child Elements of commIf Element

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the COMM_IF section. (See Section 4.3.1.6.)	Optional
2.	commIfInfo	Describes the contents of the COMM_IF_INFO part.	Required

5.3.1 COMM_IF_INFO part

5.3.1.1 Overview

A COMM_IF_INFO part describes communication interface identification information and information related to communication specifications.

Figure 5-10 shows the elements that make up the COMM_IF_INFO part when the communication interface is CC-Link. Figure 5-11 shows the elements that make up the COMM_IF_INFO part when the communication interface is CC-Link IE Field Network. Figure 5-12 shows the elements that make up the COMM_IF_INFO part when the communication interface is Ethernet. Figure 5-13 shows the elements that make up the COMM_IF_INFO part when the communication interface is CC-Link IE TSN. In the case of CC-Link IE Field Network, elements written as "When a safety communication function is available", only write in the case of the safety communication function support unit. In the case of CC-Link IE TSN, elements written as "When a safety communication function is available", only write in the case of the safety communication function support unit.

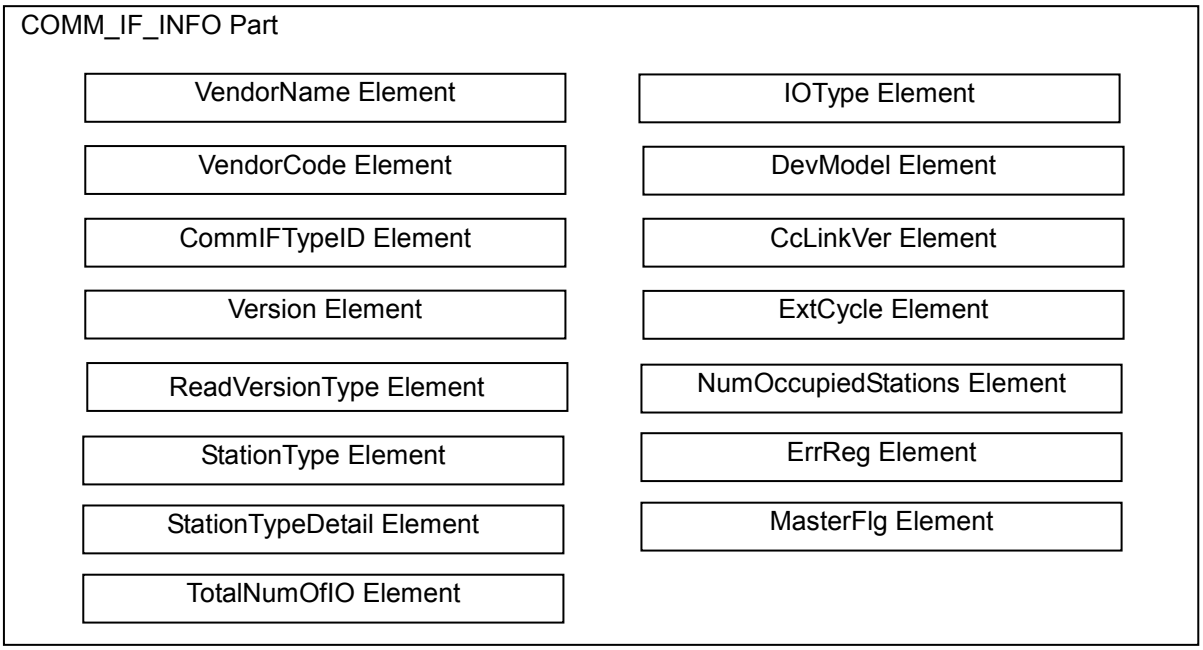


Figure 5-10 Structure of COMM_IF_INFO parts (in the case of CC-Link)

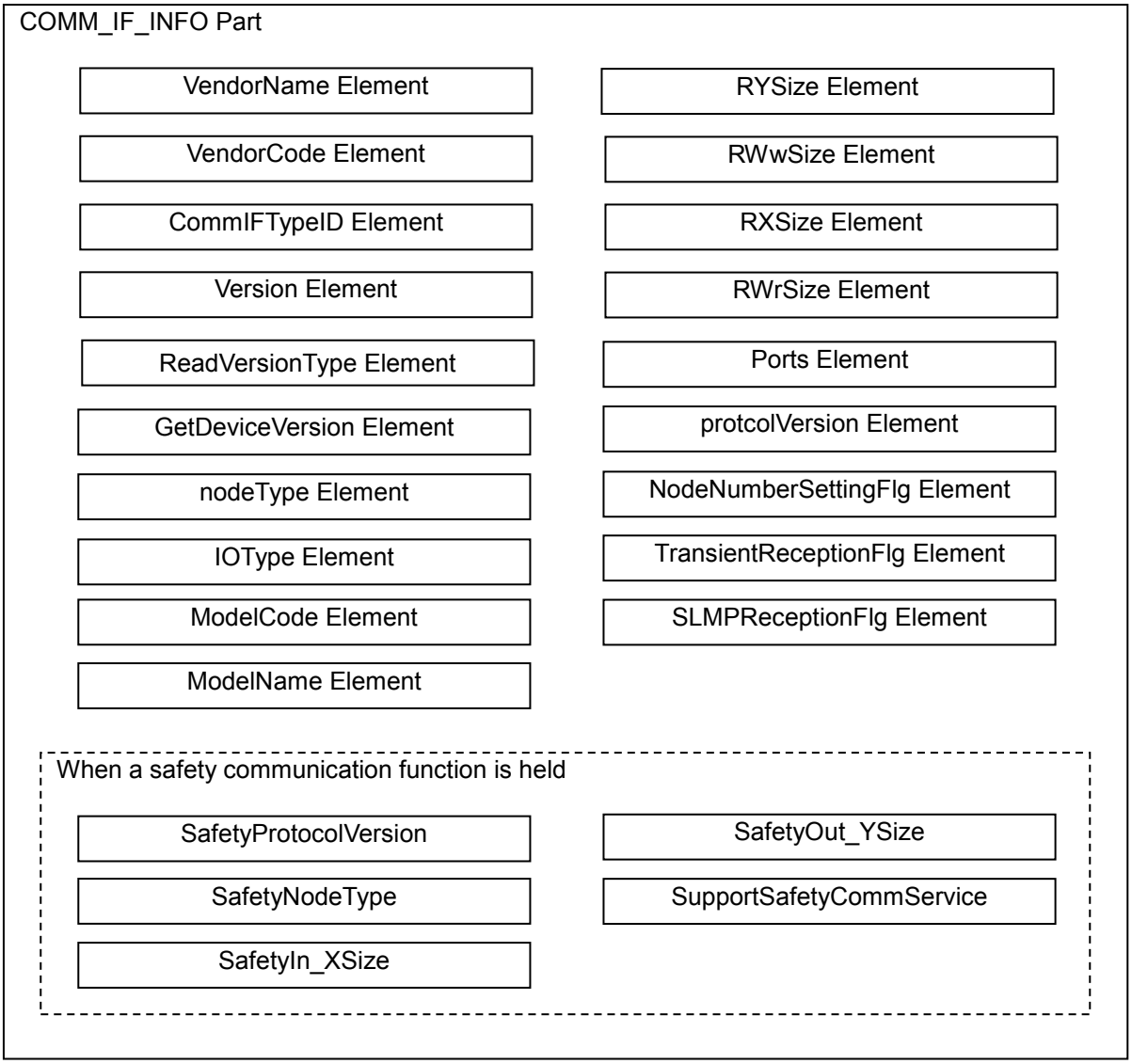


Figure 5-11 Structure of COMM_IF_INFO Part (in the case of CC-Link IE Field Network)

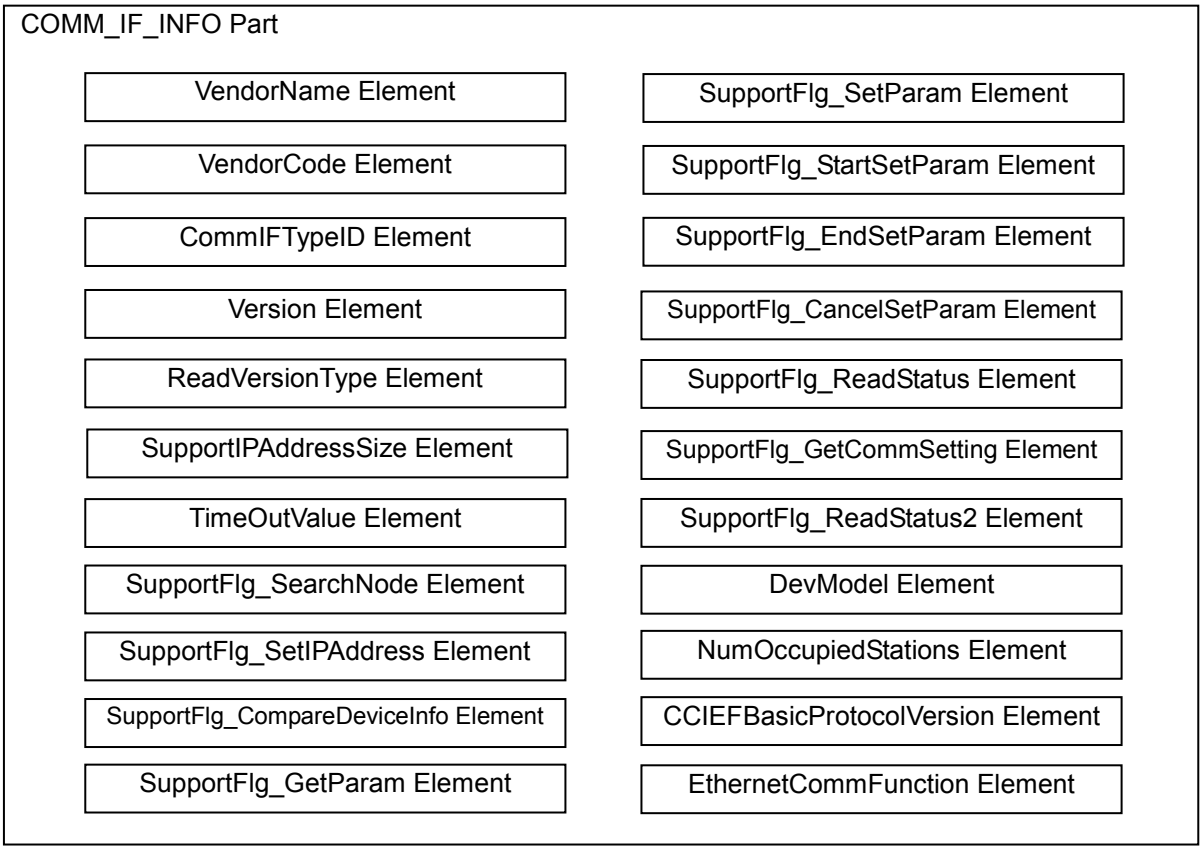


Figure 5-12 Structure of COMM_IF_INFO Parts (in the case of Ethernet)

COMM_IF_INFO part

VendorName element	R_W_DefaultSize element
VendorCode element	L_B_DefaultSize element
CommIFTypeID element	L_W_DefaultSize element
Version element	S_B_MaxSize element
StationMode element	S_W_MaxSize element
StationModeName element	R_B_MaxSize element
LocalFunction element	R_W_MaxSize element
AutoSettingHeaderType element	L_B_MaxSize element
AutoSettingType element	L_W_MaxSize element
ModelCode element	S_B_MinSize element
ModelCodeEx element	S_W_MinSize element
DevModel element	R_B_MinSize element
CanProfileNum element	R_W_MinSize element
ObjectDictionaryFileName element	L_B_MinSize element
CommunicationCycleMinTime_ns element	L_W_MinSize element
CommunicationCycleMinTime_s element	S_B_Address element
IEEE802_1ASFunction element	S_W_Address element
ReceiveFunction100M element	R_B_Address element
RelayFunction100M element	R_W_Address element
ReceiveFunction1G element	StsW_Address element
RelayFunction1G element	PDOConfigIndex+X element
MultiCastFunction element	PDOConfigPDOType+X element
CertificationClass element	PDOConfigMemoryAddress+X element
S_B_DefaultSize element	PDOConfigPossibleMapping+X element
S_W_DefaultSize element	S_General_Address element
R_B_DefaultSize element	R_General_Address element
When the safety communication function of CC-Link IE TSN is available (For details, see Figure 5-14)	

Figure 5-13 Structure of COMM_IF_INFO Part (CC-Link IE TSN)

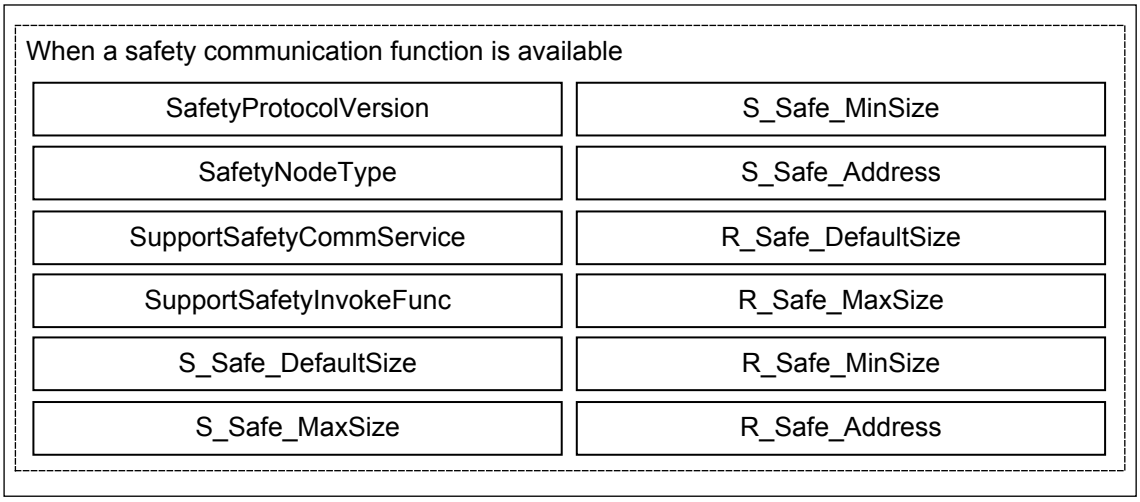


Figure 5-14 Detailed Structure of When a Safety Communication Function of CC-Link IE TSN is Available

The configuration of each element of the COMM_IF_INFO part, that is, the items that should be described within each element, is the same. Figure 5-15 shows the configuration of an element of the COMM_IF_INFO part.

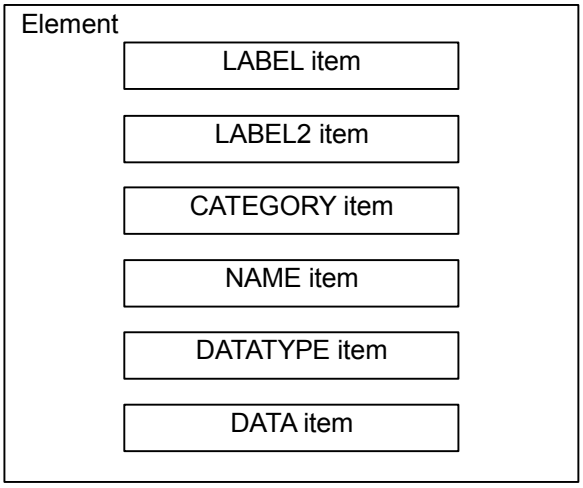


Figure 5-15 Structure Definition for Element of COMM_IF_INFO Part

Table 5-66 lists the items to be described in the elements of the COMM_IF_INFO part when the communication interface is CC-Link. A single element is described as a single XML element. For details of the description specifications of each element, see Section 5.3.1.3.

Table 5-66 Elements of COMM_IF_INFO Part with CC-Link

	No.	Element Name	Description Details	Required/Optional
Common Area	1.	VendorName	Describes the name of the vendor that manufactured the module.	Required
	2.	VendorCode	Describes the vendor code of the vendor that manufactured the module.	Required
	3.	CommIFTypeID	Describes the ID that indicates the communication interface type using a string.	Required
	4.	Version	Describes the firmware version using a string.	Required
	5.	ReadVersionType	Describes the device version acquisition method of the module.	Required
Network Dependent Area	6.	StationType	Describes the station type.	Required
	7.	StationTypeDetail	Not written when the station type is 0: Remote I/O station or 1: Remote device station. Describes the station type details when the station type is 2: Intelligent device station or local station.	See left.
	8.	TotalNumOfIO	Describes the total number of I/O points.	Optional
	9.	IOType	Describes the I/O type.	Required
	10	DevModel	Describes the model.	Required
	11.	CcLinkVer	Describes the CC-Link version.	Required
	12.	ExtCycle	Not written when the CC-Link version is earlier than 2.00. Describes the extended cyclic settings when the CC-Link version is 2.00 or later.	See left.
	13.	NumOccupiedStations	Describes the number of occupied stations.	Required
	14.	ErrReg	Describes the error code storage register.	Optional
	15.	MasterFlg	Describes whether the module can become the standby master or not.	Optional

Table 5-67 shows the items described in the elements of the COMM_IF_INFO part when the communication interface is the CC-Link IE Field Network. A single element is described as a single XML element. For details of the description specifications of each element, see Section 5.3.1.3.

Table 5-67 Elements of COMM_IF_INFO Part with CC-Link IE Field Network

	No.	Element Name	Description Details	Required/Optional
Common Area	1.	VendorName	Describes the name of the vendor that manufactured the module	Required
	2.	VendorCode	Describes the vendor code of the vendor that manufactured the module	Required
	3.	CommIFTypeID	Describes the ID that indicates the communication interface type using a string	Required
	4.	Version	Describes the firmware version using a string	Required
	5.	ReadVersionType	Describes the device version acquisition method of the module	Required
Network Dependent Area	6.	VendorName2	Describes the vendor name	Optional
	7.	nodeType	Describes the node type	Required
	8.	IOType	Describes the I/O type	Required
	9.	ModelCode	Describes the model code	Required
	10	DevModel	Describes the model	Required
	11.	ModelName	Describes the model acquirable from the module	Optional
	12.	RYSize	Describes the RY size	Required
	13.	RWwSize	Describes the RWw size	Required
	14.	RXSize	Describes the RX size	Required
	15.	RWrSize	Describes the RWr size	Required
	16.	Ports	Describes the number of ports	Required
	17.	protcolVersion	Describes the protocol version	Required
	18.	NodeNumberSetting	Describes whether or not there is a node number setting function	Required
	19.	TransientReception	Describes whether or not there is a transient reception function	Required
	20.	SLMPReception	Describes whether or not there is an SLMP reception function	Required
Function Dependent Division (Safety Communication Function)	21.	SafetyProtocolVersion	Write the CC-Link IE Safety Communication Function protocol version.	*1
	22.	SafetyNodeType	Write the safety station type.	*1
	23.	SafetyIn_XSize	Write the size of safety input data.	*1
	24.	SafetyOut_YSize	Write the size of safety output data.	*1
	25.	SupportSafetyComm Service	Write supported safety communication service. (optional)	*2

*1: Required during safety communication function support. Prohibited when not supported.
*2: Optional during safety communication function support. Prohibited when not supported.

When the communication interface is Ethernet, show the items written with the elements of COMM_IF_INFO parts in Table 5-61. Write each element as one XML element. For details of description specifications of elements, see Section 5.3.1.3.

Table 5-68 Elements of COMM_IF_INFO parts in Ethernet

	No.	Element Name	Description Details	Required /Optional
Common Area	1.	VendorName	Writes the name of the vendor that produced the unit.	Required
	2.	VendorCode	Writes the vendor code of the vendor that produced the unit.	Required
	3.	CommIFTypeID	Writes a string that shows the communication interface type.	Required
	4.	Version	Write the machine version of the unit.	Required
	5.	ReadVersionType	Writes the machine version acquisition method of the unit.	Required
Network Dependent Area	6.	SupportIPAddressSize	Writes the byte size of the IP address.	Optional
	7.	TimeOutValue	Writes the timeout value.	Optional
	8.	SupportFlg_SearchNode	Writes whether SLMPCommand "SearchNode" is supported or not.	Optional
	9.	SupportFlg_SetIPAddress	Writes whether SLMPCommand "SetIPAddress" is supported or not.	Optional
	10.	SupportFlg_CompareDeviceInfo	Writes whether SLMPCommand "CompareDeviceInformation" is supported or not.	Optional
	11.	SupportFlg_GetParam	Writes whether SLMPCommand "GetParameter" is supported or not.	Optional
	12.	SupportFlg_SetParam	Writes whether SLMPCommand "SetParameter" is supported or not.	Optional
	13.	SupportFlg_StartSetParam	Writes whether SLMPCommand "StartSetParameter" is supported or not.	Optional
	14.	SupportFlg_EndSetParam	Writes whether SLMPCommand "EndSetParameter" is supported or not.	Optional
	15.	SupportFlg_CancelSetParam	Writes whether SLMPCommand "CancelSetParameter" is supported or not.	Optional
	16.	SupportFlg_ReadStatus	Writes whether SLMPCommand "ReadStatus" is supported or not.	Optional
	17.	SupportFlg_GetCommSetting	Writes whether SLMPCommand "GetCommunicationSetting" is supported or not.	Optional
	18.	SupportFlg_ReadStatus2	Writes whether SLMPCommand "ReadStatus2" is supported or not.	Optional
	19.	DevModel	Writes the model.	*1
	20.	NumOccupiedStations	Writes the number of occupied stations.	*1
	21.	CCIEFBasicProtocolVersion	Writes the protocol version of CC-Link IE Field Network Basic.	Optional
	22.	EthernetCommFunction	Writes whether or not there is a communication function supported by a Ethernet device.	*2

*1: Required in corresponding to CC-Link IE Field Network Basic. Prohibited when not supported.
*2: Required for the CC-Link IE Field Basic communication function

Table 5-69 shows the items described in the elements of the COMM_IF_INFO part when the communication interface is the CC-Link IE TSN. Write one element as one XML element. For details of the description specifications of each element, see Section 5.3.1.3.

Table 5-69 Elements of COMM_IF_INFO Part with CC-Link IE TSN

	No.	Element Name	Description Details	Required /Optional
Common Area	1.	VendorName	Write the name of the vendor that manufactured the module.	Required
	2.	VendorCode	Write the vendor code of the vendor that manufactured the module.	Required
	3.	CommIFTypeID	Write the ID that indicates the communication interface type using a string.	Required
	4.	Version	Write the firmware version using a string.	Required
Network Dependent Area	5.	StationMode	Write the station mode setting value.	Optional
	6.	StationModeName	Write the station mode name.	*1
	7.	LocalFunction	Write whether or not there is a local function.	Optional
	8.	AutoSettingHeaderType	Write the automatic setting header type number.	Optional
	9.	AutoSettingType	Write the automatic setting type.	Optional
	10.	ModelCode	Write the model code.	Required
	11.	ModelCodeEx	Write the extended model code.	Optional
	12.	DevModel	Write the model.	Required
	13.	CanProfileNum	Write the supported CiA standard number.	Optional
	14.	ObjectDictionaryFileName	Write the object dictionary file name.	*2
	15.	CommunicationCycleMinTime_ns	Write the minimum communication cycle that can be supported by devices in the unit of ns.	*3
	16.	CommunicationCycleMinTime_s	Write the minimum communication cycle that can be supported by devices in the unit of s.	*3
	17.	IEEE802_1ASFunction	Write whether the IEEE802.1AS function is available or not.	Required
	18.	ReceiveFunction100M	Write whether or not there is a 100Mbps receive function.	Required
	19.	RelayFunction100M	Write whether or not there is a 100Mbps relay function.	Required
	20.	ReceiveFunction1G	Write whether or not there is a 1Gbps full-rate receive function.	Required
	21.	RelayFunction1G	Write whether or not there is a 1Gbps full-rate relay function.	Required
	22.	MultiCastFunction	Write whether or not there is a broadcast/multicast function.	Required
	23.	CertificationClass	Write the certification class.	Required
	24.	S_B_DefaultSize	Write the default-size send bit data.	Required
	25.	S_W_DefaultSize	Write the default-size send word data.	Required
	26.	R_B_DefaultSize	Write the default-size receive bit data.	Required
	27.	R_W_DefaultSize	Write the default-size receive word data.	Required
	28.	L_B_DefaultSize	Write the default-size link relay.	Optional
	29.	L_W_DefaultSize	Write the default-size link register.	Optional
	30.	S_B_MaxSize	Write the maximum size send bit data.	Required
	31.	S_W_MaxSize	Write the maximum size send word data.	Required
	32.	R_B_MaxSize	Write the maximum size receive bit data.	Required
	33.	R_W_MaxSize	Write the maximum size receive word data.	Required
	34.	L_B_MaxSize	Write the maximum size link relay.	Optional
	35.	L_W_MaxSize	Write the maximum size link register.	Optional
	36.	S_B_MinSize	Write the minimum size send bit data.	Optional
	37.	S_W_MinSize	Write the minimum size send word data.	Optional
	38.	R_B_MinSize	Write the minimum size receive bit data.	Optional
	39.	R_W_MinSize	Write the minimum size receive word data.	Optional

	No.	Element Name	Description Details	Required /Optional
	40.	L_B_MinSize	Write the minimum size link relay.	Optional
	41.	L_W_MinSize	Write the minimum size link register.	Optional
	42.	S_B_Address	Write the start address used for send bit data communication.	*4
	43.	S_W_Address	Write the start address used for send word data communication.	*4
	44.	R_B_Address	Write the memory address used for receive bit data communication.	*4
	45.	R_W_Address	Write the memory address used for receive word data communication.	*4
	46.	StsW_Address	Write the memory address for status notification device.	Required
	47.	PDOConfigIndex+X*6	Write the index of the PDO configuration.	*5
	48.	PDOConfigPDOType+X*6	Write the PDO type of the PDO configuration.	*6
	49.	PDOConfigMemoryAddress+X*6	Write the memory address of the PDO configuration.	*6
	50.	PDOConfigPossibleMapping+X*6	Write the mapping of the PDO configuration.	*6
	51.	S_General_Address	Write the memory address for general send access.	*4
	52.	R_General_Address	Write the memory address for general receive access.	*4
Function Dependent Area (Safety Communication Function)	53.	SafetyProtocolVersion	Write whether the CC-Link IE TSN safety communication function protocol version exists or not.	Optional
	54.	SafetyNodeType	Write the safety node type.	*9
	55.	SupportSafetyCommService	Write the service of the supported safety communication layer.	*8
	56.	SupportSafetyInvokeFunc	Write the supported safety function.	*8
	57.	S_Safe_DefaultSize	Write the default-size safety send bit data.	*9
	58.	S_Safe_MaxSize	Write the maximum size safety send bit data.	*9
	59.	S_Safe_MinSize	Write the minimum size safety send bit data.	*9
	60.	S_Safe_Address	Write the start address to be stored in the sub payload for safety send data communication.	*10
	61.	R_Safe_DefaultSize	Write the default-size safety receive bit data.	*9
	62.	R_Safe_MaxSize	Write the maximum size safety receive bit data.	*9
	63.	R_Safe_MinSize	Write the minimum size safety receive bit data.	*9
	64.	R_Safe_Address	Write the start address of the sub payload for safety receive data communication.	*11

*1: Write prohibited when StationMode is not written. When StationMode is written, "StationModeName" is required.

*2: The CAN compatible device (device with CanProfileNum written) is required. Prohibited when not supported.

*3: For the devices whose minimum communication cycle cannot be defined or the devices that do not require checking the minimum communication cycle, this may be omitted.

"CommunicationCycleMinTime_s" may be omitted for the devices whose minimum communication cycle is less than 1 second.

"CommunicationCycleMinTime_ns" may be omitted for the devices whose minimum communication cycle is represented in seconds ("CommunicationCycleMinTime_ns" is 0).

*4: The memory address information of the available communication type is required. Prohibited to write the memory addresses of the communication types for which communication is not possible.

*5: Devices for performing PDO communication are required. Write the number of indexes required. Prohibited when not supported.

*6: Be sure to write the corresponding elements when PDOConfigIndex is written. Prohibited when PDOConfigIndex is not written.
*7: Write elements that are written by entering numeric values for X as follows.
• Write elements that have the same numeric value in the order shown in Table 5-69, and be sure to write them consecutively.
• The elements must be written using consecutive numbers, and omitting (skipping) a number is prohibited.
• Write the elements so that they will be in ascending order.
*8: Prohibited for a device other than a safety compatible device (device without SafetyProtocolVersion written).
*9: Required for a safety compatible device (device with SafetyProtocolVersion written). Otherwise, writing is prohibited.
*10: Required in the case of devices with the safety send function (device of which the maximum size of the safety send data is 1 or larger). Otherwise, writing is prohibited.
*11: Required in the case of devices with the safety receive function (device of which the maximum size of the safety receive data is 1 or larger). Otherwise, writing is prohibited.

Table 5-70 shows the items described in a COMM_IF_INFO element.

Table 5-70 Item List for COMM_IF_INFO Part

No.	Item Name	Description Details	Description Specifications	Required /Optional
1.	LABEL	Describes the label for identifying the element	See Section 4.3.1.19	Required
2.	LABEL2	Describes the label for identifying the element (Used when the utility software supports another language.)	See Section 4.3.1.20	Optional
3.	CATEGORY	Describes the category for grouping the element	See Section 4.3.1.4	Optional
4.	NAME	Describes the name of the element. Used when the name and contents of the element are displayed with the utility software.	See Section 4.3.1.24	Optional
5.	DATATYPE	Describes the data type of the contents described in the DATA item.	See Section 4.3.1.10	Required
6.	DATA	Describes the contents of the element.	See Section 4.3.1.9	Required

(1) Use rules of the extended model code (ModelCodeEx)

The extended model code should be installed in the device according to the following specifications first, then described in the CSP+ file.

[Purpose]

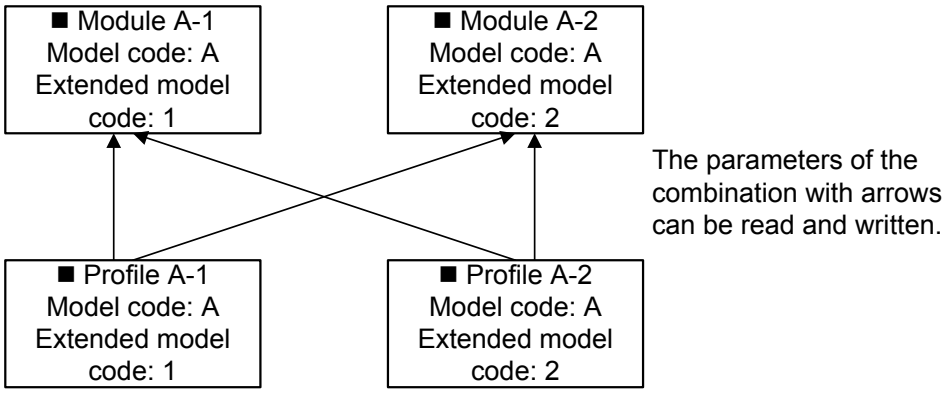
When developing a compatible device that takes over the existing module functions, use a extended model name to change the model name (* with same model code) and to make its engineering tool (such as reading/writing the parameter) compatible (to use a newly developed device with the CSP+ file of the existing module).

[Extended model code specifications]

The code starts from the number "zero" and the host devices have a larger number. The upper limit is 65535.

(a) Reading/writing the parameter

No restrictions. The parameter can be read and written in all combinations.

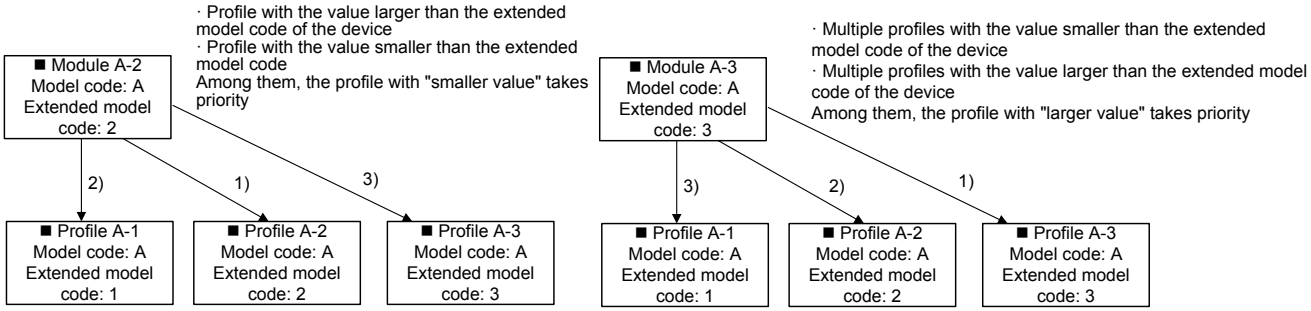


(b) Automatic detection

If there is the profile with the same extended model code, it is displayed.

If not, the profile of other extended model name with the same model code is searched.

The priority order is determined based on the following.



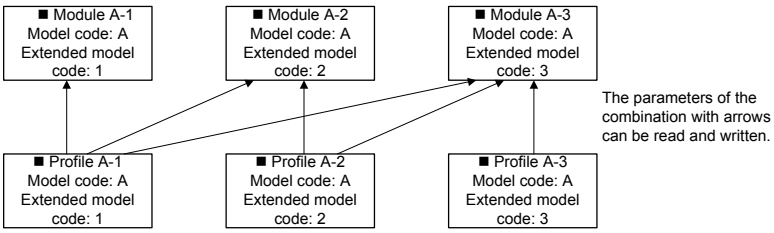
[Operation rule]

(a) The parameter can be normally read/written from the old device profile (profile with a small extended model code).

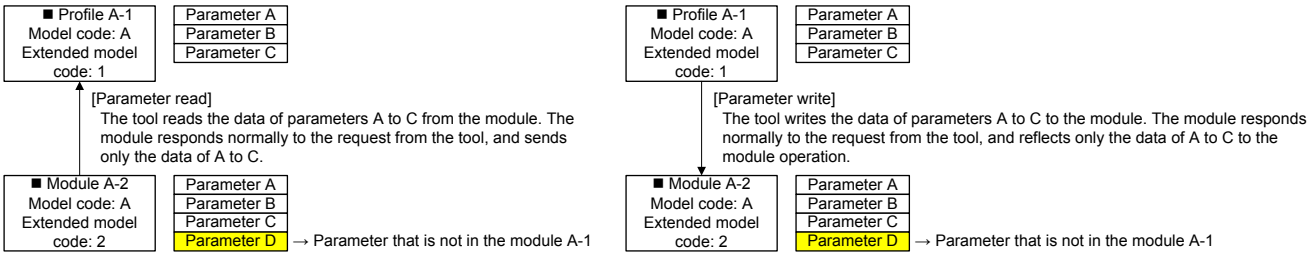
Normal: The parameters that can be set in the old device without an error can be read/written in the same way.

While reading, the parameter added with the new device is ignored.

While writing, the parameter added with the new device should be set completely without an error even though it is not written.



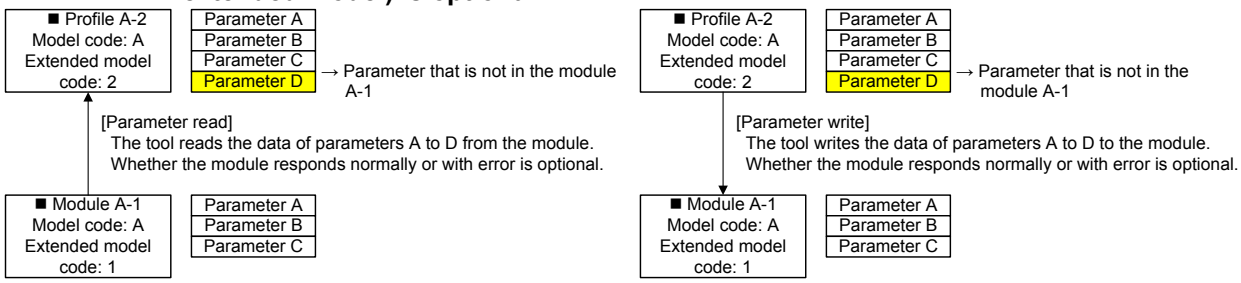
The following shows the example of parameter reading/writing.



<Reason>

It is assumed that the old device went out of production and the device should be replaced with a new device, although functions of the new device are not necessary and just the same operation as the old device is required. (Same applies to the diagnostics window)
When the new device uses a different profile, the profile needs to be updated, which bothers customers. When the new device and old device have the same model profiles, the modules cannot be distinguished on the engineering tool, especially the remote module cannot be checked with the module, which is also inconvenient for customers.
The specification "extended model code" will solve those problems.

(b) Reading/writing the parameter from the new device profile (profile with a large extended model) is optional.



The following operations are the examples.

- The module responds to an error and notifies the user that no profile corresponds to the module model.
- Only the parameters that normally response and can be recognized (parameters A to C in the above example) are read and written.
  "0" is returned for the data that cannot be recognized while reading.
  The data that cannot be recognized while writing is discarded after being received.
  However, the reserved area should be made in advance for this specifications.

(c) The contents of the network parameter should not be changed between the old and new profiles. When they are, use a device with other model code.

Contents of the network parameter: Minimum/maximum number of send cyclic points, memory address, station specific code

<Reason>

These rules are to prevent the network parameter from being changed unintendedly.
When the parameter of the A-2 unit is written to PLC with the A-2 profile and read from PLC with the personal computer that has only the A-1 profile, it changes to A-1 on the configuration window. These rules prevent the network parameter from being changed when it is written to PLC in the state.

(d) For the device version policy (VersionPolicyType), set "0: Always use the latest" or "2: Use the latest device version before the actual device version".

Also, when the extended model code is used, it is recommended not to update the device version for the operation.

<Reason>

The profile of the old extended model code and module of the new extended model code, and the profile of new extended model code and module of the old extended model code are used together respectively. If the device version is different in the combination, the combination becomes complicated.

5.3.1.2 XML format of COMM_IF_INFO part

In XML format, the COMM_IF_INFO part is described as an XML element having the XML element name "commIfInfo". Table 5-71 shows a list of the attributes of the commIfInfo element, and Table 5-72 shows a list of the XML child elements.

Table 5-71 Attribute List for commIfInfo Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the COMM_IF_INFO part. (See Section 4.3.1.19.) (Example: "CommIfInformation")	Required

Table 5-72 XML Child Element List for commIfInfo Element

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the COMM_IF_INFO part. (See Section 4.3.1.6.)	Optional
2.	commIfInfoMember	Describes the contents of the elements of the COMM_IF_INFO part.	Required

In XML format, the element of the COMM_IF_INFO part is described as an XML element having the XML element name "commIfInfoMember".

Table 5-73 shows a list of attributes of the commIfInfoMember element, and Table 5-74 shows a list of XML child elements.

Table 5-73 Attribute List for commIfInfoMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element.(See Section 4.3.1.19.) The contents written for this label are regulated by CSP+ specifications, and cannot be arbitrarily written by the user. For details, see Section 5.1.1.3.	Required

Table 5-74 XML Child Element List of commIfInfoMember Element

No.	XML Element Name	Description Details
1.	label2	Describes the LABEL2 item
2.	category	Describes the CATEGORY item
3.	name	Describes the NAME item
4.	datatype	Describes the DATATYPE item
5.	data	Describes the DATA item

Write the COMM_IF_INFO part using a format such as shown below.

```
<p:commIfInfo label="Write in accordance with the description details of Table 5-71
'Attribute Name label'''>
   <p:comment> Write in accordance with the description details of Table 5-72 'XML Element
   Name comment'</p:comment>
   <p:commIfInfoMember label="Write in accordance with the description details of Table 5-73
   'Attribute Name label'''>
      <p:label2> Write in accordance with the description details of Table 5-74 'XML Element
      Name label2'</p:label2>
      <p:category> Write in accordance with the description details of Table 5-74 'XML Element
      Name category' </p:category>
      <p:name> Write in accordance with the description details of Table 5-74  'XML Element
      Name name' </p:name>
      <p:datatype> Write in accordance with the description details of Table 5-74  'XML
      Element Name datatype'</p:datatype>
      <p:data> Write in accordance with the description details of Table 5-74 'XML Element
      Name data'</p:data>
   </p:commIfInfoMember>
   <p:commIfInfoMember label=" Write in accordance with the description details of Table
   5-73 'Attribute Name label'">
   :
   :
   </p:commIfInfoMember>
   :
   :
</p:commIfInfo>
```

5.3.1.3 Description of COMM_IF_INFO part

- **Description details of common area**

Of the elements described in the COMM_IF_INFO part, item numbers 1-5 of Table 5-59 and Table 5-67 describe common elements that remain the same regardless of changes in the communication interface. For these elements, write "COMMON" as the contents of the item name "CATEGORY".

(1) Item description of VendorName element

Table 5-75 Item Description Specifications of VendorName Element

No.	Item Name	Description Details
1.	LABEL	Write "VendorName"
2.	LABEL2	Write "VendorName" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Vendor Name" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING_U(64)"
6.	DATA	Write the name of the vendor that created the communication function **[Note]** When multiple COMM_IF sections are written in CC-Link IE TSN, the same value must be written in all the COMM_IF_INFO parts.

(a) Item description example of VendorName element

```
<p:commIfInfoMember label="VendorName">
   <p:label2><p:item>VendorName</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Vendor Name</p:item></p:name>
   <p:datatype><p:item>STRING_U(64)</p:item></p:datatype>
   <p:data><p:item>CC-Link Partner Association</p:item></p:data>
</p:commIfInfoMember>
```

(2) Item description of VendorCode element

Table 5-76 Item Description Specifications of VendorCode Element

No.	Item Name	Description Details
1.	LABEL	Write "VendorCode"
2.	LABEL2	Write "VendorCode" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Vendor Code" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "WORD"
6.	DATA	Write the code of the vendor that created the communication function Write the fifth to eighth digits of the partner member number of the CC-Link Partner Association. **[Note]** When multiple COMM_IF sections are written in CC-Link IE TSN, the same value must be written in all the COMM_IF_INFO parts.

(a) Item description example of VendorCode element

```
<p:commIfInfoMember label="VendorCode">
   <p:label2><p:item>VendorCode</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Vendor Code</p:item></p:name>
   <p:datatype><p:item>WORD</p:item></p:datatype>
   <p:data><p:item>0x0000</p:item></p:data>
</p:commIfInfoMember>
```

(3) Item description of CommIFTypeID element

Table 5-77 Item Description Specifications of CommIFTypeID Element

No.	Item Name	Description Details
1.	LABEL	Write "CommIFTypeID"
2.	LABEL2	Write "CommIFTypeID" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Communication Interface Type ID" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(32)"
6.	DATA	Write the ID that indicates the COMM_IF type Write "CC-Link" when the module supports CC-Link. Write "CCLinkIEField" when the module supports the CC-Link IE Field Network. Write "Ethernet" when the module supports the Ethernet. Write "CCLinkIETSN" for CC-Link IE TSN connection unit. **[Note]** When multiple COMM_IF sections are written in CC-Link IE TSN, the same value must be written in all the COMM_IF_INFO parts.

(a) Item description example of CommIFTypeID element

```
<p:commIfInfoMember label="CommIFTypeID">
   <p:label2><p:item>CommIFTypeID</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Communication Interface Type ID</p:item></p:name>
   <p:datatype><p:item>STRING(32)</p:item></p:datatype>
   <p:data><p:item>CCLink</p:item></p:data>
</p:commIfInfoMember>
```

(4) Item description of Version element

Table 5-78 Item Description Specifications of Version Element

No.	Item Name	Description Details
1.	LABEL	Write "Version"
2.	LABEL2	Write "Version" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Version" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "STRING(4)" when the module supports CC-Link. Select from "UINT8" and "STRING(4)" when the module supports the CC-Link IE Field Network. In the case of Ethernet connection unit, write "WORD". Write "UINT16" when the module supports CC-Link IE TSN.
6.	DATA	When the module supports CC-Link Write the value in accordance with the software version of the CC-Link Partner Association: CC-Link Specification (Overview/Protocol) BAP-C2001-001. Write A, B, C . . ., which indicates the meaning, and not 1, 2, 3 . . of the mass production device. Write "NONE" when the module does not have a software version. When the module supports the CC-Link IE Field Network Write the value equivalent to the fwVersion of the CC-Link IE Field Network Specification (Device Profile) BAP-C2005-007. In this case, the DATATYPE item needs to be "UINT8". Write "NONE" when the module does not have a FW version. In the example below, "A" is written. In the case of Ethernet connection unit, write the machine version of the target unit (the same value of the Version Element of DEVICE_INFO Part) When the module supports CC-Link IE TSN Write the value equivalent to the Device Version of Network information of the CC-Link IE TSN Specifications (Overview) BAP-C2011ENG-001. **[Note]** When multiple COMM_IF sections are written in CC-Link IE TSN, the same value must be written in all the COMM_IF_INFO parts.

(a) Item description example of Version element

```
<p:commIfInfoMember label="Version">
   <p:label2><p:item>Version</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Version</p:item></p:name>
   <p:datatype><p:item>STRING(32)</p:item></p:datatype>
   <p:data><p:item>A</p:item></p:data>
</p:commIfInfoMember>
```

(5) Item description of ReadVersionType element

Table 5-79 Item Description Specifications of ReadVersionType Element

No.	Item Name	Description Details
1.	LABEL	Write "ReadVersionType"
2.	LABEL2	Write "ReadVersionType" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Device Version Acquisition Method" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(128)"
6.	DATA	Describe the means for acquiring the module version (Version element of DEVICE_INFO part) via CC-Link. ・ Write "NA" when device version comparison is not required or the device version cannot be acquired. ・ Write "SoftwareVersion" when the "software version" acquired based on CC-Link standard specifications is the device version. ・ Write "NetworkFWVersion" when the "network firmware version" acquired based on CC-Link IE Field Network standard specifications is the device version. ・ Write "ControllerFWVersion" when the "controller firmware version" acquired based on CC-Link IE Field Network standard specifications is the device version. ・ In the case of Ethernet connection unit, write "MachineVersion". ・ Not used in the case of CC-Link IE TSN connection unit. ・ Write a reference to the METHOD or MESSAGE part using the conventions of Section 4.3.3(3) when there is a method or message for version acquisition. [Example: $(MethodSec.GetVersion)] Note that we recommend that the method/message used for reference be set as follows: ・ Set the WRITE_DATA item and REQUEST_DATA item as fixed values (do not use a VALUE reference). ・ Set only one reference to VALUE described in the READ_DATA item and RESPONSE_DATA item, and indicate the device version value to be acquired by that VALUE.

(a) Item description example of ReadVersionType element

```
<p:commIfInfoMember label="ReadVersionType">
   <p:label2><p:item>ReadVersionType</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Device Version Acquisition Method</p:item></p:name>
   <p:datatype><p:item>STRING(128)</p:item></p:datatype>
   <p:data><p:item>$(MethodSec.GetVersion)</p:item></p:data>
</p:commIfInfoMember>
```

- **Description of COMM_IF_INFO part when communication interface is CC-Link**
  Of the elements described in the COMM_IF_INFO part, item numbers 6-15 of Table 5-59 are elements described when the communication interface is CC-Link. For these elements, write "COMMON_CC-Link" as the contents of the item name "CATEGORY".

(6) Item description of StationType element

Table 5-80 Item Description Specifications of StationType Element

No.	Item Name	Description Details
1.	LABEL	Write "StationType"
2.	LABEL2	Write "StationType" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link"
4.	NAME	Write "Station Type" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "UINT16"
6.	DATA	Write the station type. 0: Remote I/O station 1: Remote device station 2: Intelligent device station or local station

(a) Item description example of StationType element

```
<p:commIfInfoMember label="StationType">
   <p:label2><p:item>StationType</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link</p:item></p:category>
   <p:name><p:item>Station Type</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>2</p:item></p:data>
</p:commIfInfoMember>
```

(7) Item description of StationTypeDetail element

Table 5-81 Item Description Specifications of StationTypeDetail Element

No.	Item Name	Description Details
1.	LABEL	Write "StationTypeDetail"
2.	LABEL2	Write "StationTypeDetail" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link"
4.	NAME	Write "Station Type Detail" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "UINT16"
6.	DATA	Write the details of the station type when the station type is [2]. When the station type is [2] and the station type details are omitted, the details are regarded as "0: Intelligent device station." 0: Intelligent device station 1: Local station

(a) Item description example of StationTypeDetail element

```
<p:commIfInfoMember label="StationTypeDetail">
   <p:label2><p:item>StationTypeDetail</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link</p:item></p:category>
   <p:name><p:item>Station Type Detail</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>0</p:item></p:data>
</p:commIfInfoMember>
```

(8) Item description of TotalNumOfIO element

Table 5-82 Item Description Specifications of TotalNumOfIO Element

No.	Item Name	Description Details
1.	LABEL	Write "TotalNumOfIO"
2.	LABEL2	Write "TotalNumOfIO" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link"
4.	NAME	Write "Total IO Points" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "UINT16"
6.	DATA	Write the total number of I/O points. 0: Depends on number of occupied stations 1: 8 points 2: 32 points 3: 16 points

(a) Item description example of TotalNumOfIO element

```
<p:commIfInfoMember label="TotalNumOfIO">
   <p:label2><p:item>TotalNumOfIO</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link</p:item></p:category>
   <p:name><p:item>Total IO Points</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>0</p:item></p:data>
</p:commIfInfoMember>
```

(9) Item description of IOType element

Table 5-83 Item Description Specifications of IOType Element

No.	Item Name	Description Details
1.	LABEL	Write "IOType"
2.	LABEL2	Write "IOType" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link"
4.	NAME	Write "IO Type" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "UINT16"
6.	DATA	Write the I/O type. 0: Front/Back mixture 1: Input 2: Output 3: Mixed

(a) Item description example of IOType element

```
<p:commIfInfoMember label="IOType">
   <p:label2><p:item>IOType</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link</p:item></p:category>
   <p:name><p:item>IO Type</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>0</p:item></p:data>
</p:commIfInfoMember>
```

(10) Item description of DevModel element

Table 5-84 Item Description Specifications of DevModel Element

No.	Item Name	Description Details
1.	LABEL	Write "DevModel"
2.	LABEL2	Write "DevModel" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link"
4.	NAME	Write "Model Name" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(48)"
6.	DATA	Write the model using a string. Write the model of the communication module when the module type is communication module.

(a) Item description example of DevModel element

```
<p:commIfInfoMember label="DevModel">
   <p:label2><p:item>DevModel</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link</p:item></p:category>
   <p:name><p:item>Model Name</p:item></p:name>
   <p:datatype><p:item>STRING(48)</p:item></p:datatype>
   <p:data><p:item>AJ65BT-D75P2-S3</p:item></p:data>
</p:commIfInfoMember>
```

(11) Item description of CcLinkVer element

Table 5-85 Item Description Specifications of CcLinkVer Element

No.	Item Name	Description Details
1.	LABEL	Write "CcLinkVer"
2.	LABEL2	Write "CcLinkVer" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link"
4.	NAME	Write "CC-Link Version" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(7)"
6.	DATA	Write the CC-Link version Write any of the following values: 1.00, 1.10, 1.11, 2.00.

(a) Item description example of CcLinkVer element

```
<p:commIfInfoMember label="CcLinkVer">
   <p:label2><p:item>CcLinkVer</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link</p:item></p:category>
   <p:name><p:item>CC-Link Version</p:item></p:name>
   <p:datatype><p:item>STRING(7)</p:item></p:datatype>
   <p:data><p:item>1.10</p:item></p:data>
</p:commIfInfoMember>
```

(12) Item description of ExtCycle element

Table 5-86 Item Description Specifications of ExtCycle Element

No.	Item Name	Description Details
1.	LABEL	Write "ExtCycle"
2.	LABEL2	Write "ExtCycle" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link"
4.	NAME	Write "Extended Cyclic Setting" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "UINT(16)"
6.	DATA	Write the extended cyclic setting when the CC-Link version is [2.00]. Write any of the following values: 1, 2, 4, 8.

(a) Item description example of ExtCycle element

```
<p:commIfInfoMember label="ExtCycle">
   <p:label2><p:item>ExtCycle</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link</p:item></p:category>
   <p:name><p:item>Extended Cyclic Setting</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(13) Item description of NumOccupiedStations element

Table 5-87 Item Description Specifications of NumOccupiedStations Element

No.	Item Name	Description Details
1.	LABEL	Write "NumOccupiedStations"
2.	LABEL2	Write "NumOccupiedStations" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link"
4.	NAME	Write "Number of Occupied Stations" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "UINT16"
6.	DATA	Write the number of occupied stations. Write any of the following values: 1, 2, 3, 4.

(a) Item description example of NumOccupiedStations element

```
<p:commIfInfoMember label="NumOccupiedStations">
   <p:label2><p:item>NumOccupiedStations</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link</p:item></p:category>
   <p:name><p:item>Number of Occupied Stations</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>4</p:item></p:data>
</p:commIfInfoMember>
```

(14) Item description of ErrReg element

Table 5-88 Item Description Specifications of ErrReg Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrReg"
2.	LABEL2	Write "ErrReg" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link"
4.	NAME	Write "Error Code Storage Register" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(7)"
6.	DATA	Write the error code storage register

(a) Item description example of ErrReg element

```
<p:commIfInfoMember label="ErrReg">
   <p:label2><p:item>ErrReg</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link</p:item></p:category>
   <p:name><p:item>Error Code Storage Register</p:item></p:name>
   <p:datatype><p:item>STRING(7)</p:item></p:datatype>
   <p:data><p:item>RWr1</p:item></p:data>
</p:commIfInfoMember>
```

(15) Item description of MasterFlg element

Table 5-89 Item Description Specifications of MasterFlg Element

No.	Item Name	Description Details
1.	LABEL	Write "MasterFlg"
2.	LABEL2	Write "MasterFlg" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link"
4.	NAME	Write "Standby Master Flag" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "UINT16"
6.	DATA	Write whether the module can become a standby master when the station type is local. 0: Cannot become standby master 1: Can become standby master

(a) Item description example of MasterFlg element

```
<p:commIfInfoMember label="MasterFlg">
   <p:label2><p:item>MasterFlg</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link</p:item></p:category>
   <p:name><p:item>Standby Master Flag</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>0</p:item></p:data>
</p:commIfInfoMember>
```

- **Description of COMM_IF_INFO part when communication interface is CC-Link IE Field Network**
  Of the elements described in the COMM_IF_INFO part, item numbers 6-19 of Table 5-67 are elements described when the communication interface is CC-Link IE Field Network. For these elements, write "COMMON_CC-Link_IE_Field" as the contents of the item name "CATEGORY".

(16) Item description of VendorName2 element

Table 5-90 Item Description Specifications of VendorName2 Element

No.	Item Name	Description Details
1.	LABEL	Write "VendorName2"
2.	LABEL2	Write "VendorName2" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field"
4.	NAME	Write "Vendor Name 2" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(32)"
6.	DATA	Write the name of the vendor that created the communication function. Write the value corresponding to VendorName of the CC-Link IE Field Network Specification (Device Profile) BAP-C2005-007. (1) This may be omitted when the description is the same as VendorName.

(a) Item description example of VendorName2 element

```
<p:commIfInfoMember label="VendorName2">
   <p:label2><p:item>VendorName2</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field</p:item></p:category>
   <p:name><p:item>Vendor Name 2</p:item></p:name>
   <p:datatype><p:item>STRING(32)</p:item></p:datatype>
   <p:data><p:item>MELSEC</p:item></p:data>
</p:commIfInfoMember>
```

(17) Item description of nodeType element

Table 5-91 Item Description Specifications of nodeType Element

No.	Item Name	Description Details
1.	LABEL	Write "nodeType"
2.	LABEL2	Write "nodeType" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field"
4.	NAME	Write "Node Type" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "BYTE"
6.	DATA	Write the node type. Write the value corresponding to nodeType of the CC-Link IE Field Network Specification (Device Profile) BAP-C2005-007. 0x32: Local station 0x33: Intelligent device station 0x34: Remote device station 0x35: Remote I/O station

(a) Item description example of nodeType element

```
<p:commIfInfoMember label="nodeType">
   <p:label2><p:item>nodeType</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field</p:item></p:category>
   <p:name><p:item>Node Type</p:item></p:name>
   <p:datatype><p:item>BYTE</p:item></p:datatype>
   <p:data><p:item>0x34</p:item></p:data>
</p:commIfInfoMember>
```

(18) Item description of IOType element

Table 5-92 Item Description Specifications of IOType Element

No.	Item Name	Description Details
1.	LABEL	Write "IOType"
2.	LABEL2	Write "IOType" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-link_IE_Field"
4.	NAME	Write "IO Type" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "UINT(16)"
6.	DATA	Write the I/O type. Write the value corresponding to ioType in the CC-Link IE Field Network Specification (Device Profile) BAP-C2005-007. 0: Front/Back mixture 1: Input 2: Output 3: Mixed

(a) Item description example of IOType element

```
<p:commIfInfoMember label="IOType">
   <p:label2><p:item>IOType</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field</p:item></p:category>
   <p:name><p:item>IO Type</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>0</p:item></p:data>
</p:commIfInfoMember>
```

(19) Item description of ModelCode element

Table 5-93 Item Description Specifications of ModelCode Element

No.	Item Name	Description Details
1.	LABEL	Write "ModelCode"
2.	LABEL2	Write "ModelCode" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field"
4.	NAME	Write "Model Code" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "UINT32"
6.	DATA	Write the model code. Write the value corresponding to modelCode of the CC-Link IE Field Network Specification (Device Profile) BAP-C2005-007. Write the model code of the communication module when the module type is communication module.

(a) Item description example of ModelCode element

```
<p:commIfInfoMember label="ModelCode">
   <p:label2><p:item>ModelCode</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field</p:item></p:category>
   <p:name><p:item>Model Code</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>1234</p:item></p:data>
</p:commIfInfoMember>
```

(20) Item description of DevModel element

Table 5-94 Item Description Specifications of DevModel Element

No.	Item Name	Description Details
1.	LABEL	Write "DevModel"
2.	LABEL2	Write "DevModel" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field"
4.	NAME	Write "Model Name" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(48)"
6.	DATA	Write the model name (model). Write the model name (model) of the communication module when the module type is communication module.

(a) Item description example of DevModel element

```
<p:commIfInfoMember label="DevModel">
   <p:label2><p:item>DevModel</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field</p:item></p:category>
   <p:name><p:item>Model Name</p:item></p:name>
   <p:datatype><p:item>STRING(48)</p:item></p:datatype>
   <p:data>
      <p:item>CC-LinkIEFieldEthernetAdapter</p:item>
   </p:data>
</p:commIfInfoMember>
```

(21) Item description of ModelName element

Table 5-95 Item Description Specifications of ModelName Element

No.	Item Name	Description Details
1.	LABEL	Write "ModelName"
2.	LABEL2	Write "ModelName" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field"
4.	NAME	Write "Model Name" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(20)"
6.	DATA	Write the model name acquirable from the module. Write the value corresponding to modelName of the CC-Link IE Field Network Specification (Device Profile) BAP-C2005-007. Write the model name of the communication module when the module type is communication module. This may be omitted when the value is the same as DevModel.

(a) Item description example of ModelName element

```
<p:commIfInfoMember label="ModelName">
   <p:label2><p:item>ModelName</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field</p:item></p:category>
   <p:name><p:item>Model Name</p:item></p:name>
   <p:datatype><p:item>STRING(20)</p:item></p:datatype>
   <p:data><p:item>EthernetAdapter</p:item></p:data>
</p:commIfInfoMember>
```

(22) Item description of RYSize element

Table 5-96 Item Description Specifications of RYSize Element

No.	Item Name	Description Details
1.	LABEL	Write "RYSize"
2.	LABEL2	Write "RYSize" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field"
4.	NAME	Write "RY Size" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "UINT16"
6.	DATA	Write the size of RY of the module. Write the value corresponding to rySize of the CC-Link Field Network Specification (Device Profile) BAP-C2005-007.

(a) Item description example of RYSize element

```
<p:commIfInfoMember label="RYSize">
   <p:label2><p:item>RYSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field</p:item></p:category>
   <p:name><p:item>RY Size</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>1024</p:item></p:data>
</p:commIfInfoMember>
```

(23) Item description of RWwSize element

Table 5-97 Item Description Specifications of RWwSize Element

No.	Item Name	Description Details
1.	LABEL	Write "RWwSize"
2.	LABEL2	Write "RWwSize" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field"
4.	NAME	Write "RWw Size" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "UINT16"
6.	DATA	Write the size of RWw of the module. Write the value corresponding to rwwSize of the CC-Link Field Network Specification (Device Profile) BAP-C2005-007.

(a) Item description example of RWwSize element

```
<p:commIfInfoMember label="RWwSize">
   <p:label2><p:item>RWwSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field</p:item></p:category>
   <p:name><p:item>RWw Size</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>1024</p:item></p:data>
</p:commIfInfoMember>
```

(24) Item description of RXSize element

Table 5-98 Item Description Specifications of RXSize Element

No.	Item Name	Description Details
1.	LABEL	Write "RXSize"
2.	LABEL2	Write the "RXSize" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field"
4.	NAME	Write the "RX Size" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "UINT16"
6.	DATA	Write the size of RX of the module. Write the value corresponding to rxSize of the CC-Link Field Network Specification (Device Profile) BAP-C2005-007.

(a) Item description example of RXSize element

```
<p:commIfInfoMember label="RXSize">
   <p:label2><p:item>RXSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field</p:item></p:category>
   <p:name><p:item>RX Size</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>1024</p:item></p:data>
</p:commIfInfoMember>
```

(25) Item description of RWrSize element

Table 5-99 Item Description Specifications of RWrSize Element

No.	Item Name	Description Details
1.	LABEL	Write "RWrSize"
2.	LABEL2	Write "RWrSize" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field"
4.	NAME	Write "RWr Size" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "UINT16"
6.	DATA	Write the size of RWr of the module. Write the value corresponding to rwrSize of the CC-Link Field Network Specification (Device Profile) BAP-C2005-007.

(a) Item description example of RWrSize element

```
<p:commIfInfoMember label="RWrSize">
   <p:label2><p:item>RWrSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field</p:item></p:category>
   <p:name><p:item>RWr Size</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>1024</p:item></p:data>
</p:commIfInfoMember>
```

(26) Item description of Ports element

Table 5-100 Item Description Specifications of Ports Element

No.	Item Name	Description Details
1.	LABEL	Write "Ports"
2.	LABEL2	Write "Ports" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field"
4.	NAME	Write "Number of Ports" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "UINT16"
6.	DATA	Write the number of ports of the module. Write the value corresponding to Ports of the CC-Link IE Field Network Specification (Device Profile) BAP-C2005-007.

(a) Item description example of Ports element

```
<p:commIfInfoMember label="Ports">
   <p:label2><p:item>Ports</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field</p:item></p:category>
   <p:name><p:item>Number of Ports</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>0</p:item></p:data>
</p:commIfInfoMember>
```

(27) Item description of protcolVersion element

Table 5-101 Item Description Specifications of protcolVersion Element

No.	Item Name	Description Details
1.	LABEL	Write "protcolVersion"
2.	LABEL2	Write "protcolVersion" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field"
4.	NAME	Write "Protcol Version" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "STRING(32)"
6.	DATA	Write the protocol version. Write the value corresponding to protocolVersion of the CC-Link Field Network Specification (Device Profile) BAP-C2005-007.

(a) Item description example of protcolVersion element

```
<p:commIfInfoMember label="protcolVersion">
   <p:label2><p:item>protcolVersion</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field</p:item></p:category>
   <p:name><p:item>Protcol Version</p:item></p:name>
   <p:datatype><p:item>STRING(32)</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(28) Item description of NodeNumberSettingFig element

Table 5-102 Item Description Specifications of NodeNumberSettingFig Element

No.	Item Name	Description Details
1.	LABEL	Write "NodeNumberSettingFig"
2.	LABEL2	Write "NodeNumberSettingFig" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field"
4.	NAME	Write "Node Number Setting Valid or Invalid" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "BOOL"
6.	DATA	Write whether or not there is a node number setting function available. Write the value corresponding to availableFuncs of the CC-Link IE Field Network Specification (Application Layer Protocol). BAP-C2005-004. 0: Function not available 1: Function available

(a) Item description example of NodeNumberSettingFig element

```
<p:commIfInfoMember label="NodeNumberSettingFig">
   <p:label2><p:item>NodeNumberSettingFig</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field</p:item></p:category>
   <p:name><p:item>Node Number Setting Valid or Invalid</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>0</p:item></p:data>
</p:commIfInfoMember>
```

(29) Item description of TransientReceptionFig element

Table 5-103 Item Description Specifications of TransientReceptionFig Element

No.	Item Name	Description Details
1.	LABEL	Write "TransientReceptionFig"
2.	LABEL2	Write "TransientReceptionFig" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field"
4.	NAME	Write "Transient Receive Function Valid or Invalid" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "BOOL"
6.	DATA	Write whether or not the transient reception function is available. Write the value corresponding to availableFuncs of the CC-Link IE Field Network Specification (Application Layer Protocol) BAP-C2005-004. 0: Function not available 1: Function available

(a) Item description example of TransientReceptionFig element

```
<p:commIfInfoMember label="TransientReceptionFig">
   <p:label2><p:item>TransientReceptionFig</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field</p:item></p:category>
   <p:name><p:item>Transient Receive Function Valid or Invalid</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>0</p:item></p:data>
</p:commIfInfoMember>
```

(30) Item description of SMLPReceptionFig element

Table 5-104 Item Description Specifications of SMLPReceptionFig Element

No.	Item Name	Description Details
1.	LABEL	Write "SMLPReceptionFig"
2.	LABEL2	Write "SMLPReception" in the case of English CSP+ files [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field"
4.	NAME	Write "SLMP Receive Function Valid or Invalid" in the case of English CSP+ files [Recommended specifications]
5.	DATATYPE	Write "BOOL"
6.	DATA	Write whether or not the SLMP reception function is available. 0: Function not available 1: Function available

(a) Item description example of SLMPReceptionFig element

```
<p:commIfInfoMember label="SLMPReceptionFig">
   <p:label2><p:item>SLMPReception</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field</p:item></p:category>
   <p:name><p:item>SLMP Receive Function Valid or Invalid</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>0</p:item></p:data>
</p:commIfInfoMember>
```

- **Details of safety communication function dependent unit**

  Of elements written in COMM_IF_INFO, this defines elements dependent on safety communication functions.

  Of elements written in COMM_IF_INFO, items 21 to 25 of Table 5-60 are the elements written when the communication interface is CC-Link IE Field Network. Write "COMMON_CC-Link_IE_Field_Safety" in the content of the "CATEGORY" item of these elements. Only write these elements in the case of the safety communication function support unit of the CC-Link IE Field Network. Safety communication functions conform to the specifications of CC-Link IE Field safety communication functions, so see "CC-Link IE Safety Communication Function Specifications (Application Layer Service and Protocol) BAP-C2007ENG-002, (Communication Profile) BAP-C2007ENG-003".

(31) Item descriptions of SafetyProtocolVersion element

Table 5-105 Item description specifications of SafetyProtocolVersion element

No.	Item Name	Description Details
1.	LABEL	Write "SafetyProtocolVersion".
2.	LABEL2	Write "SafetyProtocolVersion" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field_Safety". [Recommended Specifications]
4.	NAME	Write "Safety Communication Function protocol version" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "BYTE".
6.	DATA	Write the safety communication function protocol version. 0x00, 0x01, …, 0xFF For the safety communication function protocol version, see "CC-Link IE Safety Communication Function Specifications (Application Layer Service and Protocol) BAP-C2007ENG-002, (Communication Profile) BAP-C2007ENG-003".

(a) Example of item descriptions of SafetyProtocolVersion element

```
<p:commIfInfoMember label="SafetyProtcolVersion">
   <p:label2><p:item>SafetyProtcolVersion</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field_Safety</p:item></p:category>
   <p:name><p:item>Safety Communication Function protocol version</p:item></p:name>
   <p:datatype><p:item>BYTE</p:item></p:datatype>
   <p:data><p:item>0x02</p:item></p:data>
</p:commIfInfoMember>
```

(32) Item descriptions of SafetyNodeType element

Table 5-106 Item description specifications of SafetyNodeType element

No.	Item Name	Description Details
1.	LABEL	Write "SafetyNodeType".
2.	LABEL2	Write "SafetyNodeType" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field_Safety". [Recommended Specifications]
4.	NAME	Write "Safety Station Type" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "BYTE".
6.	DATA	Write "Safety Node Type" with the following values. 0x03: Safety intelligent device 0x04: Safety Remote Device 0x05: Safety Remote I/O See "CC-Link IE Safety Communication Function Specification (Application Layer Service and Protocol) BAP-C2007ENG-002, (Communication Profile) BAP-C2007ENG-003", IESM-SetSafetyStationInfo.

(a) Item descriptions of SafetyNodeType element

```
<p:commIfInfoMember label=" SafetyNodeType ">
   <p:label2><p:item>SafetyNodeType</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field_Safety</p:item></p:category>
   <p:name><p:item>Safety Station Type</p:item></p:name>
   <p:datatype><p:item>BYTE</p:item></p:datatype>
   <p:data><p:item>0x05</p:item></p:data>
</p:commIfInfoMember>
```

(33) Item descriptions of SafetyIn_XSize element

Table 5-107 Item description specifications of SafetyIn_XSize element

No.	Item Name	Description Details
1.	LABEL	Write "SafetyIn_XSize".
2.	LABEL2	Write "SafetyIn_XSize" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field_Safety". [Recommended Specifications]
4.	NAME	Write "Size of safety input data" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "BYTE".
6.	DATA	Write the size of safety input data.

(a) Item descriptions of SafetyIn_XSize element

```
<p:commIfInfoMember label=" SafetyIn_XSize ">
   <p:label2><p:item>SafetyIn_XSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field_Safety</p:item></p:category>
   <p:name><p:item>Size of safety input data</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>16</p:item></p:data>
</p:commIfInfoMember>
```

(34) Item descriptions of SafetyOut_YSize element

Table 5-108 Item description specifications of SafetyOut_YSize element

No.	Item Name	Description Details
1.	LABEL	Write "SafetyOut_YSize".
2.	LABEL2	Write "SafetyOut_YSize" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field_Safety". [Recommended Specifications]
4.	NAME	Write "Size of safety output data" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	Write the size of safety output data.

(a) Example of item descriptions of SafetyOut_YSize element

```
<p:commIfInfoMember label=" SafetyOut_YSize ">
   <p:label2><p:item>SafetyOut_YSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field_Safety</p:item></p:category>
   <p:name><p:item>Size of safety output data</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>32</p:item></p:data>
</p:commIfInfoMember>
```

(35) Item descriptions of SupportSafetyCommService element

Table 5-109 Item description specifications of SupportSafetyCommService element

No.	Item Name	Description Details
1.	LABEL	Write "SupportSafetyCommService".
2.	LABEL2	Write "SupportSafetyCommService" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_Field_Safety". [Recommended Specifications]
4.	NAME	Write "Supported safety communication service" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "DWORD".
6.	DATA	Each bit of the DWORD value displays the subject service, so turn ON the bit which shows the service supported by the device. (See Table 5-102 for bit and service correspondence). Write the value when turned ON of bits for services not supported. If not even one service shown in Table 5-102 is not supported, either omit it or write 0x00000000. (If omitted, it will be counted as 0x00000000) When showing that there is need to support multiple services, write the value of multiple bits turned ON.

(a) Example of item descriptions of SupportSafetyCommService element

```
<p:commIfInfoMember label="SupportSafetyCommService">
   <p:label2><p:item>SupportSafetyCommService</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_Field_Safety</p:item></p:category>
   <p:name><p:item>Supported safety communication service</p:item></p:name>
   <p:datatype><p:item>DWORD</p:item></p:datatype>
   <p:data><p:item>0x00000001</p:item></p:data>
</p:commIfInfoMember>
```

Table 5-110 Services shown by each bit

Bit Position	Service
0	IESS-Start
1	IESS-InvokeFunc
2 to 31	For future expansion (Set to 0).

For safety communication service, see "CC-Link IE Safety Communication Function Specification (Application Layer Service and Protocol) (Communication Profile) BAP-C2007ENG-002, BAP-C2007ENG-003".

- **Description of COMM_IF_INFO Part when the communication interface is Ethernet**
  Of elements written in COMM_IF_INFO Part, items 5 to 18 of Table 5-61 are the elements written when the communication interface is Ethernet. Write "COMMON_Ethernet" in the content of the "CATEGORY" item of these elements.

(36) Item descriptions of SupportIPAddressSize element

Either omit this element, or if it is written, set the value of the DATA item to "4".

Table 5-111 Item description specifications of SupportIPAddressSize

No.	Item Name	Description Details
1.	LABEL	Write "SupportIPAddressSize".
2.	LABEL2	Write "SupportIPAddressSize" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_Ethernet". [Recommended Specifications]
4.	NAME	Write "Support IP Address Size" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	Write The byte size of the IP address supported in set type notation. "4" (IPv4), "16" (IPv6) or both of those listed can be considered for the value of this element, but for the time being it will only support 4 (IPv4). If this element is omitted, it is considered in the same manner as "4".

(a) Example of item descriptions of SupportIPAddressSize element

```
<p:commIfInfoMember label="SupportIPAddressSize">
   <p:label2><p:item>SupportIPAddressSize</p:item></p:label2>
   <p:category><p:item>COMMON_Ethernet</p:item></p:category>
   <p:name><p:item>Support IP Address Size</p:item></p:name>
   <p:datatype><p:item>UINT16()</p:item></p:datatype>
   <p:data><p:item>4</p:item></p:data>
</p:commIfInfoMember>
```

(37) Item descriptions of TimeOutValue element

Table 5-112 Item description specifications of TimeOutValue element

No.	Item Name	Description Details
1.	LABEL	Write "TimeOutValue".
2.	LABEL2	Write "TimeOutValue" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_Ethernet". [Recommended Specifications]
4.	NAME	Write "Time Out Value"" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	IF the IP address of the unit can be set in SLMP, write the time out value of the Set IP Address command in integers of millisecond units. If this element is omitted, it is considered in the same manner as if "30000" (30 seconds) was entered.

(a) Example of item description of TimeOutValue element

```
<p:commIfInfoMember label="TimeOutValue">
   <p:label2><p:item>TimeOutValue</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>Time Out Value</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>100</p:item></p:data>
</p:commIfInfoMember>
```

(38) Item descriptions of SupportFlg_SearchNode

Table 5-113 Item description specifications of SupportFlg_SearchNode

No.	Item Name	Description Details
1.	LABEL	Write "SupportFlg_SearchNode".
2.	LABEL2	Write "SupportFlg_SearchNode" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_Ethernet". [Recommended Specifications]
4.	NAME	Write "SupportFlg_SearchNode" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether the "SearchNode" SLMP command is supported or not. 0: Not supported 1: Supported If this element is not written, it shall be considered the same as if "0" was written.

(a) Example of item descriptions of SupportFlg_SearchNode

```
<p:commIfInfoMember label="SupportFlg_SearchNode">
   <p:label2><p:item>SupportFlg_ SearchNode</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>SupportFlg_ SearchNode</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(39) Item descriptions of SupportFlg_SetIPAddress element

Table 5-114 Item description specifications of SupportFlg_SetIPAddress element

No.	Item Name	Description Details
1.	LABEL	Write "SupportFlg_SetIPAddress".
2.	LABEL2	Write "SupportFlg_SetIPAddress" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_Ethernet". [Recommended Specifications]
4.	NAME	Write "SupportFlg_SetIPAddress" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether the "SetIPAddress" SLMP command is supported or not. 0: Not supported 1: Supported If this element is not written, it shall be considered the same as if "0" was written.

(a) Example of item description of SupportFlg_SetIPAddress element

```
<p:commIfInfoMember label="SupportFlg_SetIPAddress">
   <p:label2><p:item>SupportFlg_SetIPAddress</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>SupportFlg_SetIPAddress</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(40) Item descriptions of SupportFlg_CompareDeviceInfo

Table 5-115 Item description specifications of SupportFlg_CompareDeviceInfo

No.	Item Name	Description Details
1.	LABEL	Write "SupportFlg_CompareDeviceInfo".
2.	LABEL2	Write "SupportFlg_CompareDeviceInfo" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_Ethernet". [Recommended Specifications]
4.	NAME	Write "SupportFlg_CompareDeviceInfo" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether the "CompareDeviceInfo" SLMP command is supported or not. 0: Not supported 1: Supported If this element is not written, it shall be considered the same as if "0" was written.

(a) Example of item descriptions of SupportFlg_CompareDeviceInfo

```
<p:commIfInfoMember label="SupportFlg_CompareDeviceInfo">
   <p:label2><p:item>SupportFlg_CompareDeviceInfo</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>SupportFlg_CompareDeviceInfo</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(41) Item descriptions of SupportFlg_GetParam element

Table 5-116 Item description specifications of SupportFlg_GetParam element

No.	Item Name	Description Details
1.	LABEL	Write "SupportFlg_GetParam".
2.	LABEL2	Write "SupportFlg_GetParam" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_Ethernet". [Recommended Specifications]
4.	NAME	Write "SupportFlg_GetParam" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether the "GetParameter" SLMP command is supported or not. 0: Not supported 1: Supported If this element is not written, it shall be considered the same as if "0" was written.

(a) Example of item description of SupportFlg_GetParam element

```
<p:commIfInfoMember label="SupportFlg_GetParam">
   <p:label2><p:item>SupportFlg_GetParam</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>SupportFlg_GetParam</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(42) Item descriptions of SupportFlg_CompareDeviceInfo

Table 5-117 Item description specifications of SupportFlg_SetParam

No.	Item Name	Description Details
1.	LABEL	Write "SupportFlg_SetParam".
2.	LABEL2	Write "SupportFlg_SetParam" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_Ethernet". [Recommended Specifications]
4.	NAME	Write "SupportFlg_SetParam" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether the "SetParameter" SLMP command is supported or not. 0: Not supported 1: Supported If this element is not written, it shall be considered the same as if "0" was written.

(a) Example of item descriptions of SupportFlg_SetParam

```
<p:commIfInfoMember label="SupportFlg_SetParam">
   <p:label2><p:item>SupportFlg_SetParam</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>SupportFlg_SetParam</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(43) Item descriptions of SupportFlg_StartSetParam element

Table 5-118 Item description specifications of SupportFlg_StartSetParam element

No.	Item Name	Description Details
1.	LABEL	Write "SupportFlg_StartSetParam".
2.	LABEL2	Write "SupportFlg_StartSetParam" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_Ethernet". [Recommended Specifications]
4.	NAME	Write "SupportFlg_StartSetParam" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether the "StartSetParameter" SLMP command is supported or not. 0: Not supported 1: Supported If this element is not written, it shall be considered the same as if "0" was written.

(a) Example of item description of SupportFlg_StartSetParam element

```
<p:commIfInfoMember label="SupportFlg_StartSetParam">
   <p:label2><p:item>SupportFlg_StartSetParam</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>SupportFlg_StartSetParam</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(44) Item descriptions of SupportFlg_EndSetParam

Table 5-119 Item description specifications of SupportFlg_EndSetParam

No.	Item Name	Description Details
1.	LABEL	Write "SupportFlg_EndSetParam".
2.	LABEL2	Write "SupportFlg_EndSetParam" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_Ethernet". [Recommended Specifications]
4.	NAME	Write "SupportFlg_EndSetParam" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether the "EndSetParameter" SLMP command is supported or not. 0: Not supported 1: Supported If this element is not written, it shall be considered the same as if "0" was written.

(a) Example of item descriptions of SupportFlg_EndSetParam

```
<p:commIfInfoMember label="SupportFlg_EndSetParam">
   <p:label2><p:item>SupportFlg_EndSetParam</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>SupportFlg_EndSetParam</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(45) Item descriptions of SupportFlg_CancelSetParam element

Table 5-120 Item description specifications of SupportFlg_CancelSetParam element

No.	Item Name	Description Details
1.	LABEL	Write "SupportFlg_CancelSetParam".
2.	LABEL2	Write "SupportFlg_CancelSetParam" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_Ethernet". [Recommended Specifications]
4.	NAME	Write "SupportFlg_CancelSetParam" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether the "CancelSetParameter" SLMP command is supported or not. 0: Not supported 1: Supported If this element is not written, it shall be considered the same as if "0" was written.

(a) Example of item description of SupportFlg_CancelSetParam element

```
<p:commIfInfoMember label="SupportFlg_CancelSetParam">
   <p:label2><p:item>SupportFlg_CancelSetParam</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>SupportFlg_CancelSetParam</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(46) Item descriptions of SupportFlg_ReadStatus

Table 5-121 Item description specifications of SupportFlg_ReadStatus

No.	Item Name	Description Details
1.	LABEL	Write "SupportFlg_ReadStatus".
2.	LABEL2	Write "SupportFlg_ReadStatus" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_Ethernet". [Recommended Specifications]
4.	NAME	Write "SupportFlg_ReadStatus" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether the "ReadStatus" SLMP command is supported or not. 0: Not supported 1: Supported If this element is not written, it shall be considered the same as if "0" was written.

(a) Example of item descriptions of SupportFlg_ReadStatus

```
<p:commIfInfoMember label="SupportFlg_ReadStatus">
   <p:label2><p:item>SupportFlg_ReadStatus</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>SupportFlg_ReadStatus</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(47) Item descriptions of SupportFlg_GetCommSetting element

Table 5-122 Item description specifications of SupportFlg_GetCommSetting element

No.	Item Name	Description Details
1.	LABEL	Write "SupportFlg_GetCommSetting".
2.	LABEL2	Write "SupportFlg_GetCommSetting" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_Ethernet". [Recommended Specifications]
4.	NAME	Write "SupportFlg_GetCommSetting" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether the "GetCommunicationSetting" SLMP command is supported or not. 0: Not supported 1: Supported If this element is not written, it shall be considered the same as if "0" was written.

(a) Example of item description of SupportFlg_GetCommSetting element

```
<p:commIfInfoMember label="SupportFlg_GetCommSetting">
   <p:label2><p:item>SupportFlg_GetCommSetting</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>SupportFlg_GetCommSetting</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(48) Item descriptions of SupportFlg_ReadStatus2 element

Table 5-123 Item description specifications of SupportFlg_ReadStatus2 element

No.	Item Name	Description Details
1.	LABEL	Write "SupportFlg_ReadStatus2".
2.	LABEL2	Write "SupportFlg_ReadStatus2" in the case of English CSP+ files. [Recommended Specifications]
3.	CATEGORY	Write "COMMON_Ethernet". [Recommended Specifications]
4.	NAME	Write "SupportFlg_ReadStatus2" in the case of English CSP+ files. [Recommended Specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether the "ReadStatus2" SLMP command is supported or not. 0: Not supported 1: Supported If this element is not written, it shall be considered the same as if "0" was written.

(a) Example of item description of SupportFlg_ReadStatus2 element

```
<p:commIfInfoMember label="SupportFlg_ReadStatus2">
   <p:label2><p:item>SupportFlg_ReadStatus2</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>SupportFlg_ReadStatus2</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(49) Item description of DevModel element

Table 5-124 Item Description Specifications of DevModel Element

No.	Item Name	Description Details
1.	LABEL	Write "DevModel".
2.	LABEL2	Write "DevModel" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_Ethernet".
4.	NAME	Write "Model Name" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "STRING(48)".
6.	DATA	Write the model using a string.

(a) Item description example of DevModel element

```
<p:commIfInfoMember label="DevModel">
   <p:label2><p:item>DevModel</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>Model Name</p:item></p:name>
   <p:datatype><p:item>STRING(48)</p:item></p:datatype>
   <p:data><p:item>DIGITALOUT-CCIESIM-16</p:item></p:data>
</p:commIfInfoMember>
```

(50) Item description of NumOccupiedStations element

Table 5-125 Item Description Specifications of NumOccupiedStations Element

No.	Item Name	Description Details
1.	LABEL	Write "NumOccupiedStations".
2.	LABEL2	Write "NumOccupiedStations" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_Ethernet".
4.	NAME	Write "Number of Occupied Stations" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	Write the number of occupied stations. Write any integer value from 1 to 64.

(a) Item description example of NumOccupiedStations element

```
<p:commIfInfoMember label="NumOccupiedStations">
   <p:label2><p:item>NumOccupiedStations</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>Number of Occupied Stations</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>2</p:item></p:data>
</p:commIfInfoMember>
```

(51) Item description of CCIEFBasicProtocolVersion element

Table 5-126 Item Description Specifications of CCIEFBasicProtocolVersion Element

No.	Item Name	Description Details
1.	LABEL	Write "CCIEFBasicProtocolVersion".
2.	LABEL2	Write "CCIEFBasicProtocolVersion" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_Ethernet".
4.	NAME	Write "CCIEF Basic Protocol Version" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	Write the protocol version of CC-Link IE Field Network Basic. If this element is omitted, it will be regarded as description of "1".

(a) Item description example of CCIEFBasicProtocolVersion element

```
<p:commIfInfoMember label="CCIEFBasicProtocolVersion">
   <p:label2><p:item>CCIEFBasicProtocolVersion</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>CCIEF Basic Protocol Version</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(52) Item description of EthernetCommFunction element

Table 5-127 Item Description Specifications of EthernetCommFunction Element

No.	Item Name	Description Details
1.	LABEL	Write "EthernetCommFunction".
2.	LABEL2	Write "EthernetCommFunction" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_Ethernet".
4.	NAME	Write "Ethernet Communication Functions" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "DWORD".
6.	DATA	Write whether or not there is a communication function supported by a Ethernet device. Express the presence or absence of one communication function per bit. The meanings of each bit value are described as follows: 0: Not functional 1: Functional Functions corresponding to each bit value are described as follows: bit0: SLMP communication function bit1: CC-Link IE Field Network Basic communication function bit2 to bit3: For future extension use (treating this as 0 in this version) If this element is omitted, it will be regarded as description of "0x00000001".

(a) Item description example of EthernetCommFunction element

```
<p:commIfInfoMember label="EthernetCommFunction">
   <p:label2><p:item>EthernetCommFunction</p:item></p:label2>
   <p:category><p:item>COMMON_ Ethernet</p:item></p:category>
   <p:name><p:item>Ethernet Communication Functions</p:item></p:name>
   <p:datatype><p:item>DWORD</p:item></p:datatype>
   <p:data><p:item>0x00000002</p:item></p:data>
</p:commIfInfoMember>
```

- **Description of COMM_IF_INFO part when communication interface is the CC-Link IE TSN**
  For the elements described in the COMM_IF_INFO part, item number 5 to 51 of Table 5-69 are the elements described when the communication interface is CC-Link IE TSN. For the item name "CATEGORY", write "COMMON_CC-Link_IE_TSN" as the contents.

(53) Item description of StationMode element

Table 5-128 Item Description Specifications of StationMode Element

No.	Item Name	Description Details
1.	LABEL	Write "StationMode".
2.	LABEL2	Write "StationMode" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Station Mode Setting Value" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "WORD".
6.	DATA	Write the station mode setting value. **[Note]** When multiple COMM_IF sections are written in CC-Link IE TSN, different values must be written in all the COMM_IF_INFO parts.

(a) Item description example of StationMode element

```
<p:commIfInfoMember label="StationMode">
   <p:label2><p:item>StationMode</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Station Mode Setting Value</p:item></p:name>
   <p:datatype><p:item>WORD</p:item></p:datatype>
   <p:data><p:item>0x0001</p:item></p:data>
</p:commIfInfoMember>
```

(54) Item description of StationModeName element

Table 5-129 Item Description Specifications of StationModeName Element

No.	Item Name	Description Details
1.	LABEL	Write "StationModeName".
2.	LABEL2	Write "StationModeName" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Station Mode Name" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "STRING_U(32)".
6.	DATA	Write the station mode name.

(a) Item description example of StationModeName element

```
<p:commIfInfoMember label="StationModeName">
   <p:label2><p:item>StationModeName</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Station Mode Name</p:item></p:name>
   <p:datatype><p:item>STRING_U(32)</p:item></p:datatype>
   <p:data><p:item>Motion Mode</p:item></p:data>
</p:commIfInfoMember>
```

(55) Item description of LocalFunction element

Table 5-130 Item Description Specifications of LocalFunction Element

No.	Item Name	Description Details
1.	LABEL	Write "LocalFunction".
2.	LABEL2	Write "LocalFunction" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Local Function" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write the local function. 0: Disabled 1: Enabled If this element is omitted, it is considered in the same manner as "0". **[Note]** When multiple COMM_IF sections are written in CC-Link IE TSN, the same value must be written in all the COMM_IF_INFO parts.

(a) Item description example of LocalFunction element

```
<p:commIfInfoMember label="LocalFunction">
   <p:label2><p:item>LocalFunction</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Local Function</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(56) Item description of AutoSettingHeaderType element

Table 5-131 Item Description Specifications of AutoSettingHeaderType Element

No.	Item Name	Description Details
1.	LABEL	Write "AutoSettingHeaderType".
2.	LABEL2	Write "AutoSettingHeaderType" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Automatic Setting Header Type" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "BYTE".
6.	DATA	Write the header type number (0x00 to 0xFF). For the header type number, see the parameter automatic setting file format in "CC-Link IE TSN Specifications (Application Layer Protocol) BAP-C2011ENG-004". If this element is not written, it shall be considered the same as if "0x00" was written.

(a) Item description example of AutoSettingHeaderType element

```
<p:commIfInfoMember label="AutoSettingHeaderType">
   <p:label2><p:item>AutoSettingHeaderType</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Automatic Setting Header Type</p:item></p:name>
   <p:datatype><p:item>BYTE</p:item></p:datatype>
   <p:data><p:item>0x00</p:item></p:data>
</p:commIfInfoMember>
```

(57) Item description of AutoSettingType element

Table 5-132 Item Description Specifications of AutoSettingType Element

No.	Item Name	Description Details
1.	LABEL	Write "AutoSettingType".
2.	LABEL2	Write "AutoSettingType" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Auto Setting Type" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "WORD".
6.	DATA	Write the automatic setting type. 0x0000: Function not available (default) 0x0001: Settable for individual parameters only 0x0002: Settable for batch parameters only If this element is not written, it shall be considered the same as if "0x0000" was written. If this element is omitted, it is considered in the same manner as "0". When 0x0001 is set, always write MESSAGE for parameter auto setting described in Section 5.5.10.2. When 0x0002 is set, parameter related information (values of METHOD_TYPE item and MESSAGE_TYPE item are elements of PARAMETER and AUTO_PARAMETER) is not supported.

(a) Item description example of AutoSettingType element

```
<p:commIfInfoMember label="AutoSettingType">
   <p:label2><p:item>AutoSettingType</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Auto Setting Type</p:item></p:name>
   <p:datatype><p:item>WORD</p:item></p:datatype>
   <p:data><p:item>0x0001</p:item></p:data>
</p:commIfInfoMember>
```

(58) Item description of ModelCode element

Table 5-133 Item Description Specifications of ModelCode Element

No.	Item Name	Description Details
1.	LABEL	Write "ModelCode".
2.	LABEL2	Write "ModelCode" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Model Code" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the model code. Write the model code of the network information of the CC-Link IE TSN Specifications (Overview) BAP-C2011ENG-001. Write the model of the communication module when the module type is communication module. **[Note]** When multiple COMM_IF sections are written in CC-Link IE TSN, the same value must be written in all the COMM_IF_INFO parts.

(a) Item description example of ModelCode element

```
<p:commIfInfoMember label="ModelCode">
   <p:label2><p:item>ModelCode</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Model Code</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>1234</p:item></p:data>
</p:commIfInfoMember>
```

(59) Item description of ModelCodeEx element

Table 5-134 Item Description Specifications of ModelCodeEx Element

No.	Item Name	Description Details
1.	LABEL	Write "ModelCodeEx".
2.	LABEL2	Write "ModelCodeEx" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Extended Model Code" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	Write the extended model code. Write the extended model code of the CC-Link IE TSN Specifications (Overview) BAP-C2011ENG-001. **[Note]** When multiple COMM_IF sections are written in CC-Link IE TSN, the same value must be written in all the COMM_IF_INFO parts. For the application rules for the extended model code, see (1) in Section 5.3.1.1.

(a) Item description example of ModelCodeEx element

```
<p:commIfInfoMember label="ModelCodeEx">
   <p:label2><p:item>ModelCodeEx</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Extended Model Code/p:item></p:name>
   <p:datatype><p:item> UINT16</p:item></p:datatype>
   <p:data><p:item>9876</p:item></p:data>
</p:commIfInfoMember>
```

(60) Item description of DevModel element

Table 5-135 Item Description Specifications of DevModel Element

No.	Item Name	Description Details
1.	LABEL	Write "DevModel".
2.	LABEL2	Write "DevModel" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Model Name" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "STRING(48)".
6.	DATA	Write the model. Write the model name of the communication module when the module type is communication module. **[Note]** When multiple COMM_IF sections are written in CC-Link IE TSN, the same value must be written in all the COMM_IF_INFO parts.

(a) Item description example of DevModel element

```
<p:commIfInfoMember label="DevModel">
   <p:label2><p:item>DevModel</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Model Name</p:item></p:name>
   <p:datatype><p:item>STRING(48)</p:item></p:datatype>
   <p:data><p:item> CC-Link IE TSNEthernetAdapter </p:item></p:data>
</p:commIfInfoMember>
```

(61) Item description of CanProfileNum element

Table 5-136 Item Description Specifications of CanProfileNum Element

No.	Item Name	Description Details
1.	LABEL	Write "CanProfileNum".
2.	LABEL2	Write "CanProfileNum" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Supported CiA Standard Number" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "STRING(32)".
6.	DATA	Write the supported CiA standard number. 401: The CiA401 standard is supported. 402: The CiA402 standard is supported. If this element is omitted, it is considered that CAN is not supported.

(a) Item description example of CanProfileNum element

```
<p:commIfInfoMember label="CanProfileNum">
   <p:label2><p:item>CanProfileNum</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Supported CiA Standard Number</p:item></p:name>
   <p:datatype><p:item>STRING(32)</p:item></p:datatype>
   <p:data><p:item>401</p:item></p:data>
</p:commIfInfoMember>
```

(62) Item description of ObjectDictionaryFileName element

Table 5-137 Item Description Specifications of ObjectDictionaryFileName Element

No.	Item Name	Description Details
1.	LABEL	Write "ObjectDictionaryFileName".
2.	LABEL2	Write the "ObjectDictionaryFileName" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write the "Object Dictionary File Name" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "STRING(52)".
6.	DATA	Write the object dictionary file name including the extension (.csv). The characters that can be used in the file name are single-byte alphanumeric characters and symbols only.

(a) Item description example of ObjectDictionaryFileName element

```
<p:commIfInfoMember label="ObjectDictionaryFileName">
   <p:label2><p:item>ObjectDictionaryFileName</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Object Dictionary File Name</p:item></p:name>
   <p:datatype><p:item>STRING(52)</p:item></p:datatype>
   <p:data><p:item>A1234.csv</p:item></p:data>
</p:commIfInfoMember>
```

(63) Item description of CommunicationCycleMinTime_ns element

Table 5-138 Item Description Specifications of CommunicationCycleMinTime_ns Element

No.	Item Name	Description Details
1.	LABEL	Write "CommunicationCycleMinTime_ns".
2.	LABEL2	Write "CommunicationCycleMinTime_ns" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "CommunicationCycleMinTime_ns" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	When using devices in the supported station mode, write the minimum communication cycle that can be supported by the devices in the unit of ns. The writable range is 0 to 999999999. [Note] When the minimum communication cycle is 1 second or longer, write it with CommunicationCycleMinTime_s as a set.

(a) Item description example of CommunicationCycleMinTime_ns element

```
<p:commIfInfoMember label=" CommunicationCycleMinTime_ns ">
        <p:label2><p:item>CommunicationCycleMinTime_ns </p:item></p:label2>
        <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
        <p:name><p:item>CommunicationCycleMinTime_ns</p:item></p:name>
        <p:datatype><p:item>UINT32</p:item></p:datatype>
        <p:data><p:item>10000</p:item></p:data>
</p:commIfInfoMember>
```

(64) Item description of CommunicationCycleMinTime_s element

Table 5-139 Item Description specifications of CommunicationCycleMinTime_s Element

No.	Item Name	Description Details
1.	LABEL	Write "CommunicationCycleMinTime_s".
2.	LABEL2	Write "CommunicationCycleMinTime_s" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "CommunicationCycleMinTime_s" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	When using devices in the supported station mode, write the minimum communication cycle that can be supported by the devices in the unit of s. The writable range is 0 to 65535.

(a) Item description example of CommunicationCycleMinTime_s element

```
<p:commIfInfoMember label="CommunicationCycleMinTime_s">
        <p:label2><p:item>CommunicationCycleMinTime_s</p:item></p:label2>
        <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
        <p:name><p:item>CommunicationCycleMinTime_s</p:item></p:name>
        <p:datatype><p:item>UINT16</p:item></p:datatype>
        <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(65) Item description of IEEE802_1ASFunction element

Table 5-140 Item Description Specifications of IEEE802_1ASFunction Element

No.	Item Name	Description Details
1.	LABEL	Write "IEEE802_1ASFunction".
2.	LABEL2	Write "IEEE802_1ASFunction" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "IEEE802_1AS Function" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether the IEEE802.1AS function is available or not. 0: Function not available 1: Function available Write whether the IEEE802.1AS function used in CC-Link IE TSN Specification (Overview) BAP-C2011ENG-001 is available or not.

(a) Item description example of IEEE802_1ASFunction element

```
<p:commIfInfoMember label="IEEE802_1ASFunction">
   <p:label2><p:item>IEEE802_1AS Function</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>IEEE802_1AS Function</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(66) Item description of ReceiveFunction100M element

Table 5-141 Item Description Specifications of ReceiveFunction100M Element

No.	Item Name	Description Details
1.	LABEL	Write "ReceiveFunction100M".
2.	LABEL2	Write "ReceiveFunction100M" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Receive Function_100Mbps" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether or not there is a 100Mbps full-rate receive function. 0: Function not available 1: Function available Write the value equivalent to the certification class information in the CC-Link IE TSN Specifications (Overview) BAP-C2011ENG-001.

(a) Item description example of ReceiveFunction100M element

```
<p:commIfInfoMember label="ReceiveFunction100M">
   <p:label2><p:item>ReceiveFunction100M</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Receive Function_100Mbps</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(67) Item description of RelayFunction100M element

Table 5-142 Item Description Specifications of RelayFunction100M Element

No.	Item Name	Description Details
1.	LABEL	Write "RelayFunction100M".
2.	LABEL2	Write "RelayFunction100M" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Relay Function_100Mbps" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether or not there is a 100Mbps full-rate relay function. 0: Function not available 1: Function available Write the value equivalent to the certification class information in the CC-Link IE TSN Specifications (Overview) BAP-C2011ENG-001.

(a) Item description example of RelayFunction100M element

```
<p:commIfInfoMember label="RelayFunction100M">
   <p:label2><p:item>RelayFunction100M</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Relay Function_100Mbps</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(68) Item description of ReceiveFunction1G element

Table 5-143 Item Description Specifications of ReceiveFunction1G Element

No.	Item Name	Description Details
1.	LABEL	Write "ReceiveFunction1G".
2.	LABEL2	Write "ReceiveFunction1G" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Receive Function_1Gbps" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether or not there is a 1Gbps full-rate receive function. 0: Function not available 1: Function available Write the value equivalent to the certification class information in the CC-Link IE TSN Specifications (Overview) BAP-C2011ENG-001.

(a) Item description example of ReceiveFunction1G element

```
<p:commIfInfoMember label="ReceiveFunction1G">
   <p:label2><p:item>ReceiveFunction1G</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Receive Function_1Gbps</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>0</p:item></p:data>
</p:commIfInfoMember>
```

(69) Item description of RelayFunction1G element

Table 5-144 Item Description Specifications of RelayFunction1G Element

No.	Item Name	Description Details
1.	LABEL	Write "RelayFunction1G".
2.	LABEL2	Write "RelayFunction1G" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Relay Function_1Gbps" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether or not there is a 1Gbps full-rate relay function. 0: Function not available 1: Function available Write the value equivalent to the certification class information in the CC-Link IE TSN Specifications (Overview) BAP-C2011ENG-001.

(a) Item description example of RelayFunction1G element

```
<p:commIfInfoMember label="RelayFunction1G">
   <p:label2><p:item>RelayFunction1G</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Relay Function_1Gbps</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(70) Item description of MultiCastFunction element

Table 5-145 Item Description Specifications of MultiCastFunction Element

No.	Item Name	Description Details
1.	LABEL	Write "MultiCastFunction".
2.	LABEL2	Write "MultiCastFunction" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Broad_MultiCast Function" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DATA	Write whether or not there is a broadcast/multicast function. 0: Function not available 1: Function available Write the value equivalent to the certification class information in the CC-Link IE TSN Specifications (Overview) BAP-C2011ENG-001.

(a) Item description example of MultiCastFunction element

```
<p:commIfInfoMember label="MultiCastFunction">
   <p:label2><p:item>MultiCastFunction</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Broad_MultiCast Function</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(71) Item description of CertificationClass element

Table 5-146 Item Description Specifications of CertificationClass Element

No.	Item Name	Description Details
1.	LABEL	Write "CertificationClass".
2.	LABEL2	Write "CertificationClass" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Certification Class" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "STRING_U(2)".
6.	DATA	Write the character indicating the certification class. Example: A Write the value equivalent to the certification class information in the CC-Link IE TSN Specifications (Overview) BAP-C2011ENG-001.

(a) Item description example of CertificationClass element

```
<p:commIfInfoMember label="CertificationClass">
   <p:label2><p:item>CertificationClass</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Certification Class</p:item></p:name>
   <p:datatype><p:item>STRING_U(2)</p:item></p:datatype>
   <p:data><p:item>A</p:item></p:data>
</p:commIfInfoMember>
```

(72) Item description of S_B_DefaultSize element

Table 5-147 Item Description Specifications of S_B_DefaultSize Element

No.	Item Name	Description Details
1.	LABEL	Write "S_B_DefaultSize".
2.	LABEL2	Write "S_B_DefaultSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Send Bit Data Default Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the default size of the send bit data (RX) in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum".

(a) Item description example of S_B_DefaultSize element

```
<p:commIfInfoMember label="S_B_DefaultSize">
   <p:label2><p:item>S_B_DefaultSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Send Bit Data Default Size</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>30</p:item></p:data>
</p:commIfInfoMember>
```

(73) Item description of S_W_DefaultSize element

Table 5-148 Item Description Specifications of S_W_DefaultSize Element

No.	Item Name	Description Details
1.	LABEL	Write "S_W_DefaultSize".
2.	LABEL2	Write "S_W_DefaultSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Send Word Data Default Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	Write the default size of the send word data (RWr, TPDO, general send access) in the number of words. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum". In the case of devices that use TPDO communication, write a value equal to or larger than the default number of TPDO mapping points.

(a) Item description example of S_W_DefaultSize element

```
<p:commIfInfoMember label="S_W_DefaultSize">
   <p:label2><p:item>S_W_DefaultSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Send Word Data Default Size</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>15</p:item></p:data>
</p:commIfInfoMember>
```

(74) Item description of R_B_DefaultSize element

Table 5-149 Item Description Specifications of R_B_DefaultSize Element

No.	Item Name	Description Details
1.	LABEL	Write "R_B_DefaultSize".
2.	LABEL2	Write "R_B_DefaultSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Receive Bit Data Default Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the default size of the receive bit data (RY) in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum".

(a) Item description example of R_B_DefaultSize element

```
<p:commIfInfoMember label="R_B_DefaultSize">
   <p:label2><p:item>R_B_DefaultSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Receive Bit Data Default Size</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>18</p:item></p:data>
</p:commIfInfoMember>
```

(75) Item description of R_W_DefaultSize element

Table 5-150 Item Description Specifications of R_W_DefaultSize Element

No.	Item Name	Description Details
1.	LABEL	Write "R_W_DefaultSize".
2.	LABEL2	Write "R_W_DefaultSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Receive Word Data Default Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	Write the default size of the receive word data (RWw, RPDO, general receive access) in the number of words. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum". In the case of devices that use RPDO communication, write a value equal to or larger than the default number of RPDO mapping points.

(a) Item description example of R_W_DefaultSize element

```
<p:commIfInfoMember label="R_W_DefaultSize">
   <p:label2><p:item>R_W_DefaultSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Receive Word Data Default Size</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>14</p:item></p:data>
</p:commIfInfoMember>
```

(76) Item description of L_B_DefaultSize element

Table 5-151 Item Description Specifications of L_B_DefaultSize Element

No.	Item Name	Description Details
1.	LABEL	Write "L_B_DefaultSize".
2.	LABEL2	Write "L_B_DefaultSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Link Relay Default Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the default size of the link relay data (LB) in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum". Prohibited for a device that does not support the local function.

(a) Item description example of L_B_DefaultSize element

```
<p:commIfInfoMember label="L_B_DefaultSize">
   <p:label2><p:item>L_B_DefaultSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Link Relay Default Size</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>30</p:item></p:data>
</p:commIfInfoMember>
```

(77) Item description of L_W_DefaultSize element

Table 5-152 Item Description Specifications of L_W_DefaultSize Element

No.	Item Name	Description Details
1.	LABEL	Write "L_W_DefaultSize".
2.	LABEL2	Write "L_W_DefaultSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Link Register Default Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	Write the default size of the link register data (LW) in the number of words. Write a numerical value other than 0. The value must be within the range "minimum ≤ default ≤ maximum". Prohibited for a device that does not support the local function.

(a) Item description example of L_W_DefaultSize element

```
<p:commIfInfoMember label="L_W_DefaultSize">
   <p:label2><p:item>L_W_DefaultSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Link Register Default Size</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>50</p:item></p:data>
</p:commIfInfoMember>
```

(78) Item description of S_B_MaxSize element

Table 5-153 Item Description Specifications of S_B_MaxSize Element

No.	Item Name	Description Details
1.	LABEL	Write "S_B_MaxSize".
2.	LABEL2	Write "S_B_MaxSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Send Bit Data Maximum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the maximum size of the send bit data (RX) in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum".

(a) Item description example of S_B_MaxSize element

```
<p:commIfInfoMember label="S_B_MaxSize">
   <p:label2><p:item>S_B_MaxSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Send Bit Data Maximum Size</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>1024</p:item></p:data>
</p:commIfInfoMember>
```

(79) Item description of S_W_MaxSize element

Table 5-154 Item Description Specifications of S_W_MaxSize Element

No.	Item Name	Description Details
1.	LABEL	Write "S_W_MaxSize".
2.	LABEL2	Write "S_W_MaxSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Send Word Data Maximum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	Write the maximum size of the send word data (RWr, TPDO, general send access) in the number of words. Write a numerical equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum". In the case of devices that use TPDO communication, write a value equal to or larger than the number of TPDO mapping points in the maximum size.

(a) Item description example of S_W_MaxSize element

```
<p:commIfInfoMember label="S_W_MaxSize">
   <p:label2><p:item>S_W_MaxSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Send Word Data Maximum Size</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>512</p:item></p:data>
</p:commIfInfoMember>
```

(80) Item description of R_B_MaxSize element

Table 5-155 Item Description Specifications of R_B_MaxSize Element

No.	Item Name	Description Details
1.	LABEL	Write "R_B_MaxSize".
2.	LABEL2	Write "R_B_MaxSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Receive Bit Data Maximum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the maximum size of the receive bit data (RY) in the number of bits. Write a numerical equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum".

(a) Item description example of R_B_MaxSize element

```
<p:commIfInfoMember label="R_B_MaxSize">
   <p:label2><p:item>R_B_MaxSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>R_B_MaxSize</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>2048</p:item></p:data>
</p:commIfInfoMember>
```

(81) Item description of R_W_MaxSize element

Table 5-156 Item Description Specifications of R_W_MaxSize Element

No.	Item Name	Description Details
1.	LABEL	Write "R_W_MaxSize".
2.	LABEL2	Write "R_W_MaxSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Receive Word Data Maximum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	Write the maximum size of the receive word data (RWw, RPDO, general receive access) in the number of words. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum". In the case of devices that use RPDO communication, write a value equal to or larger than the number of RPDO mapping points at the maximum size.

(a) Item description example of R_W_MaxSize element

```
<p:commIfInfoMember label="R_W_MaxSize">
   <p:label2><p:item>R_W_MaxSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Receive Word Data Maximum Size</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>512</p:item></p:data>
</p:commIfInfoMember>
```

(82) Item description of L_B_MaxSize element

Table 5-157 Item Description Specifications of L_B_MaxSize Element

No.	Item Name	Description Details
1.	LABEL	Write "L_B_MaxSize".
2.	LABEL2	Write "L_B_MaxSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Link Relay Maximum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the maximum size of the link relay data (LB) in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum". Prohibited for a device that does not support the local function.

(a) Item description example of L_B_MaxSize element

```
<p:commIfInfoMember label="L_B_MaxSize">
   <p:label2><p:item>L_B_MaxSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Link Relay Maximum Size</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>1024</p:item></p:data>
</p:commIfInfoMember>
```

(83) Item description of L_W_MaxSize element

Table 5-158 Item Description Specifications of L_W_MaxSize Element

No.	Item Name	Description Details
1.	LABEL	Write "L_W_MaxSize".
2.	LABEL2	Write "L_W_MaxSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Link Register Maximum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	Write the maximum size of the link register data (LW) in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum". Prohibited for a device that does not support the local function.

(a) Item description example of L_W_MaxSize element

```
<p:commIfInfoMember label="L_W_MaxSize">
   <p:label2><p:item>L_W_MaxSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Link Register Maximum Size</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>512</p:item></p:data>
</p:commIfInfoMember>
```

(84) Item description of S_B_MinSize element

Table 5-159 Item Description Specifications of S_B_MinSize Element

No.	Item Name	Description Details
1.	LABEL	Write "S_B_MinSize".
2.	LABEL2	Write "S_B_MinSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Send Bit Data Minimum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the minimum size of the send bit data (RX) in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum". If omitted, 0 is regarded to be set.

(a) Item description example of S_B_MinSize element

```
<p:commIfInfoMember label="S_B_MinSize">
   <p:label2><p:item>S_B_MinSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Send Bit Data Minimum Size</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(85) Item description of S_W_MinSize element

Table 5-160 Item Description Specifications of S_W_MinSize Element

No.	Item Name	Description Details
1.	LABEL	Write "S_W_MinSize".
2.	LABEL2	Write "S_W_MinSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Send Word Data Minimum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	Write the minimum size of the send word data (RWr, TPDO, general send access) in the number of words. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum". If omitted, 0 is regarded to be set. In the case of devices that use TPDO communication, write a value equal to or larger than the number of TPDO mapping points at the minimum size.

(a) Item description example of S_W_MinSize element

```
<p:commIfInfoMember label="S_W_MinSize">
   <p:label2><p:item>S_W_MinSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Send Word Data Minimum Size</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>0</p:item></p:data>
</p:commIfInfoMember>
```

(86) Item description of R_B_MinSize element

Table 5-161 Item Description Specifications of R_B_MinSize Element

No.	Item Name	Description Details
1.	LABEL	Write "R_B_MinSize".
2.	LABEL2	Write "R_B_MinSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Receive Bit Data Minimum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the minimum size of the receive bit data (RY) in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum". If omitted, 0 is regarded to be set.

(a) Item description example of R_B_MinSize element

```
<p:commIfInfoMember label="R_B_MinSize">
   <p:label2><p:item>R_B_MinSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Receive Bit Data Minimum Size</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>2</p:item></p:data>
</p:commIfInfoMember>
```

(87) Item description of R_W_MinSize element

Table 5-162 Item Description Specifications of R_W_MinSize Element

No.	Item Name	Description Details
1.	LABEL	Write "R_W_MinSize".
2.	LABEL2	Write "R_W_MinSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Receive Word Data Minimum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	Write the minimum size of the receive word data (RWw, RPDO, general receive access) in the number of words. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum". If omitted, 0 is regarded to be set. In the case of devices that use RPDO communication, write a value equal to or larger than the number of RPDO mapping points at the minimum size.

(a) Item description example of R_W_MinSize element

```
<p:commIfInfoMember label="R_W_MinSize">
   <p:label2><p:item>R_W_MinSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Receive Word Data Minimum Size</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(88) Item description of L_B_MinSize element

Table 5-163 Item Description Specifications of L_B_MinSize Element

No.	Item Name	Description Details
1.	LABEL	Write "L_B_MinSize".
2.	LABEL2	Write "L_B_MinSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Link Relay Minimum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the minimum size of the link relay data (LB) in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum". If omitted, 0 is regarded to be set. Prohibited for a device that does not support the local function.

(a) Item description example of L_B_MinSize element

```
<p:commIfInfoMember label="L_B_MinSize">
   <p:label2><p:item>L_B_MinSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Link Relay Minimum Size</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(89) Item description of L_W_MinSize element

Table 5-164 Item Description Specifications of L_W_MinSize Element

No.	Item Name	Description Details
1.	LABEL	Write "L_W_MinSize".
2.	LABEL2	Write "L_W_MinSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Link Register Minimum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT16".
6.	DATA	Write the minimum size of the link register data (LW) in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum". If omitted, 0 is regarded to be set. Prohibited for a device that does not support the local function.

(a) Item description example of L_W_MinSize element

```
<p:commIfInfoMember label="L_W_MinSize">
   <p:label2><p:item>L_W_MinSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Link Register Minimum Size</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(90) Item description of S_B_Address element

Table 5-165 Item Description Specifications of S_B_Address Element

No.	Item Name	Description Details
1.	LABEL	Write "S_B_Address".
2.	LABEL2	Write "S_B_Address" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Send Bit Data Address" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "DWORD".
6.	DATA	Write the start address used for send bit data (Rx) communication.

(a) Item description example of S_B_Address element

```
<p:commIfInfoMember label="S_B_Address">
   <p:label2><p:item>S_B_Address</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Send Bit Data Address</p:item></p:name>
   <p:datatype><p:item>DWORD</p:item></p:datatype>
   <p:data><p:item>0x00000123</p:item></p:data>
</p:commIfInfoMember>
```

(91) Item description of S_W_Address element

Table 5-166 Item Description Specifications of S_W_Address Element

No.	Item Name	Description Details
1.	LABEL	Write "S_W_Address".
2.	LABEL2	Write "S_W_Address" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Send Word Data Address" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "DWORD".
6.	DATA	Write the start address used for send word data (RWr) communication.

(a) Item description example of S_W_Address element

```
<p:commIfInfoMember label="S_W_Address">
   <p:label2><p:item>S_W_Address</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Send Word Data Address</p:item></p:name>
   <p:datatype><p:item>DWORD</p:item></p:datatype>
   <p:data><p:item>0x00000456</p:item></p:data>
</p:commIfInfoMember>
```

(92) Item description of R_B_Address element

Table 5-167 Item Description Specifications of R_B_Address Element

No.	Item Name	Description Details
1.	LABEL	Write "R_B_Address".
2.	LABEL2	Write "R_B_Address" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Receive Bit Data Address" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "DWORD".
6.	DATA	Write the memory address used for receive bit data (RY) communication.

(a) Item description example of R_B_Address element

```
<p:commIfInfoMember label="R_B_Address">
   <p:label2><p:item>R_B_Address</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Receive Bit Data Address</p:item></p:name>
   <p:datatype><p:item>DWORD</p:item></p:datatype>
   <p:data><p:item>0x00000789</p:item></p:data>
</p:commIfInfoMember>
```

(93) Item description of R_W_Address element

Table 5-168 Item Description Specifications of R_W_Address Element

No.	Item Name	Description Details
1.	LABEL	Write "R_W_Address".
2.	LABEL2	Write "R_W_Address" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Receive Word Data Address" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "DWORD".
6.	DATA	Write the memory address used for receive word data (RWw) communication.

(a) Item description example of R_W_Address element

```
<p:commIfInfoMember label="R_W_Address">
   <p:label2><p:item>R_W_Address</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Receive Word Data Address</p:item></p:name>
   <p:datatype><p:item>DWORD</p:item></p:datatype>
   <p:data><p:item>0x00000159</p:item></p:data>
</p:commIfInfoMember>
```

(94) Item description of StsW_Address element

Table 5-169 Item Description Specifications of StsW_Address Element

No.	Item Name	Description Details
1.	LABEL	Write "StsW_Address".
2.	LABEL2	Write "StsW_Address" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "Status Notification Device Address" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "DWORD".
6.	DATA	Write the memory address for the status notification device (StsW).

(a) Item description example of StsW_Address element

```
<p:commIfInfoMember label="StsW_Address">
   <p:label2><p:item>StsW_Address</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>Status Notification Device Address</p:item></p:name>
   <p:datatype><p:item>DWORD</p:item></p:datatype>
   <p:data><p:item>0x00000753</p:item></p:data>
</p:commIfInfoMember>
```

(95) Item description of PDOConfigIndex+X element

Table 5-170 Item Description Specifications of PDOConfigIndex+X Element

No.	Item Name	Description Details
1.	LABEL	Write "PDOConfigIndex+X". Write an integer equal to or larger than 1 for X in order. This value cannot be omitted.
2.	LABEL2	Write "PDOConfigIndex+X" in the case of English CSP+ files. [Recommended specifications] Write any integer for X.
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "PDO Configuration Index +X" in the case of English CSP+ files. [Recommended specifications] Write any integer for X.
5.	DATATYPE	Write "WORD".
6.	DATA	Write the index of the PDO configuration. Write the range referring to "D.2 PDO Config" in the CC-Link IE TSN Specifications (Overview) BAP-C2011ENG-001.

(a) Item description example of PDOConfigIndex1 element

```
<p:commIfInfoMember label="PDOConfigIndex1">
   <p:label2><p:item>PDOConfigIndex1</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>PDO Configuration Index 1</p:item></p:name>
   <p:datatype><p:item>WORD</p:item></p:datatype>
   <p:data><p:item>0x1C00</p:item></p:data>
</p:commIfInfoMember>
```

(96) Item description of PDOConfigPDOType+X element

Table 5-171 Item Description Specifications of PDOConfigPDOType+X Element

No.	Item Name	Description Details
1.	LABEL	Write "PDOConfigPDOType+X". Write the same value as PDOConfigIndex for X. This value cannot be omitted.
2.	LABEL2	Write "PDOConfigPDOType+X" in the case of English CSP+ files. [Recommended specifications] Write any integer for X.
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "PDO Configuration PDO Type +X" in the case of English CSP+ files. [Recommended specifications] Write any integer for X.
5.	DATATYPE	Write "BYTE".
6.	DATA	Write the PDO type of the PDO configuration. 0x00:RPDO 0x01:TPDO

(a) Item description example of PDOConfigPDOType1 element

```
<p:commIfInfoMember label="PDOConfigPDOType1">
   <p:label2><p:item>PDOConfigPDOType1</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>PDO Configuration PDO Type 1</p:item></p:name>
   <p:datatype><p:item>BYTE</p:item></p:datatype>
   <p:data><p:item>0x01</p:item></p:data>
</p:commIfInfoMember>
```

(97) Item description of PDOConfigMemoryAddress+X element

Table 5-172 Item Description Specifications of PDOConfigMemoryAddress+X Element

No.	Item Name	Description Details
1.	LABEL	Write "PDOConfigMemoryAddress+X". Write the same value as PDOConfigIndex for X. This value cannot be omitted.
2.	LABEL2	Write "PDOConfigMemoryAddress+X" in the case of English CSP+ files. [Recommended specifications] Write any integer for X.
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "PDO Configuration Memory Address +X" in the case of English CSP+ files. [Recommended specifications] Write any integer for X.
5.	DATATYPE	Write "DWORD".
6.	DATA	Write the memory address of the PDO configuration. For RPDO: Write the memory address of the sub payload. For TPDO: Write the start address of the data to be stored in the sub payload.

(a) Item description example of PDOConfigMemoryAddress1 element

```
<p:commIfInfoMember label="PDOConfigMemoryAddress1">
   <p:label2><p:item>PDOConfigMemoryAddress1</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>PDO Configuration Memory Address 1</p:item></p:name>
   <p:datatype><p:item>DWORD</p:item></p:datatype>
   <p:data><p:item>0x00000001</p:item></p:data>
</p:commIfInfoMember>
```

(98) Item description of PDOConfigPossibleMapping+X element

Table 5-173 Item Description Specifications of PDOConfigPossibleMapping+X Element

No.	Item Name	Description Details
1.	LABEL	Write "PDOConfigPossibleMapping+X". Write the same value as PDOConfigIndex for X. This value cannot be omitted.
2.	LABEL2	Write "PDOConfigPossibleMapping+X" in the case of English CSP+ files. [Recommended specifications] Write any integer for X.
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "PDO Configuration Mapping Candidate +X" in the case of English CSP+ files. [Recommended specifications] Write any integer for X. * Write a name for NAME in consideration of 0x00 of SubIndex of the object dictionary.
5.	DATATYPE	Write "WORD[x]". Number of x elements: 1 to 512
6.	DATA	Write the PDO mapping objects that can be set in PDO Assignment of the PDO configuration as an array in priority order. Example: <0x1600><0x1601> The first element will be the default.

(a) Item description example of PDOConfigPossibleMapping1 element

```
<p:commIfInfoMember label="PDOConfigPossibleMapping1">
   <p:label2><p:item>PDOConfigPossibleMapping1</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>PDO Configuration Mapping Candidate 1/p:item></p:name>
   <p:datatype><p:item>WORD[2]</p:item></p:datatype>
   <p:data><p:item><0x1600><0x1601></p:item></p:data>
</p:commIfInfoMember>
```

(99) Item description of S_General_Address element

Table 5-174 Item Description Specifications of S_General_Address Element

No.	Item Name	Description Details
1.	LABEL	Write "S_General_Address".
2.	LABEL2	Write "S_General_Address" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "General Send Memory Address" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "DWORD".
6.	DATA	Write the memory address for general send access.

(a) Item description example of S_General_Address element

```
<p:commIfInfoMember label="S_General_Address">
   <p:label2><p:item>S_General_Address</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>General Send Memory Address</p:item></p:name>
   <p:datatype><p:item>DWORD</p:item></p:datatype>
   <p:data><p:item>0x00000741</p:item></p:data>
</p:commIfInfoMember>
```

(100) Item description of R_General_Address element

Table 5-175 Item Description Specifications of R_General_Address Element

No.	Item Name	Description Details
1.	LABEL	Write "R_General_Address".
2.	LABEL2	Write "R_General_Address" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN".
4.	NAME	Write "General Receive Memory Address" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "DWORD".
6.	DATA	Write the memory address for general receive access.

(a) Item description example of R_General_Address element

```
<p:commIfInfoMember label="R_General_Address">
   <p:label2><p:item>R_General_Address</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN</p:item></p:category>
   <p:name><p:item>General Receive Memory Address</p:item></p:name>
   <p:datatype><p:item>DWORD</p:item></p:datatype>
   <p:data><p:item>0x00000369</p:item></p:data>
</p:commIfInfoMember>
```

- **Details of CC-Link IE TSN safety communication function dependent area**

  Of elements written in the COMM_IF_INFO part, this defines those elements that are dependent on the safety communication function. For the elements described in the COMM_IF_INFO part, item number 53 to 64 of Table 5-69 are the elements described when the communication interface is CC-Link IE TSN. For the item name "CATEGORY", write "COMMON_CC-Link_IE_TSN_Safety" as the contents. Only write these elements in the case of the safety communication function support unit of CC-Link IE TSN. The CC-Link IE TSN safety communication function conforms to the specifications of the CC-Link IE TSN safety communication function, so see "CC-Link IE Safety Communication Function Specification (Application Layer Service and Protocol) BAP-C2007ENG-002 and (Communication Profile) BAP-C2007ENG-003".

(101) Item description of SafetyProtocolVersion element

Table 5-176 Item Description Specifications of SafetyProtocolVersion Element

No.	Item Name	Description Details
1.	LABEL	Write "SafetyProtocolVersion".
2.	LABEL2	Write "SafetyProtocolVersion" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN_Safety".
4.	NAME	Write "Safety Communication Function Protocol Version" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "BYTE".
6.	DATA	Write the safety communication function protocol version. 0x00, 0x01, ...0xFF For details on the safety communication function protocol version, see the safety communication function protocol version of "CC-Link IE Safety Communication Function Specifications (Application Layer and Service Protocol) BAP-C2007ENG-002 and (Communication Profile) BAP-C2007ENG-003".

(a) Item description example of SafetyProtocolVersion element

```
<p:commIfInfoMember label="SafetyProtocolVersion">
   <p:label2><p:item>SafetyProtocolVersion</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN_Safety</p:item></p:category>
   <p:name><p:item>Safety Communication Function Protocol Version</p:item></p:name>
   <p:datatype><p:item>BYTE</p:item></p:datatype>
   <p:data><p:item>0x01</p:item></p:data>
</p:commIfInfoMember>
```

(102) Item description of SafetyNodeType element

Table 5-177 Item Description Specifications of SafetyNodeType Element

No.	Item Name	Description Details
1.	LABEL	Write "SafetyNodeType".
2.	LABEL2	Write "SafetyNodeType" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN_Safety".
4.	NAME	Write "Safety Node Type" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "BYTE".
6.	DATA	Write the safety node type. 0x02: Safety local 0x06: Safety remote

(a) Item description example of SafetyNodeType element

```
<p:commIfInfoMember label="SafetyNodeType">
   <p:label2><p:item>SafetyNodeType</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN_Safety</p:item></p:category>
   <p:name><p:item>Safety Node Type</p:item></p:name>
   <p:datatype><p:item>BYTE</p:item></p:datatype>
   <p:data><p:item>0x02</p:item></p:data>
</p:commIfInfoMember>
```

(103) Item description of SupportSafetyCommService element

Table 5-178 Item Description Specifications of SupportSafetyCommService Element

No.	Item Name	Description Details
1.	LABEL	Write "SupportSafetyCommService".
2.	LABEL2	Write "SupportSafetyCommService" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN_Safety".
4.	NAME	Write "Support Safety Communication Service" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "DWORD".
6.	DATA	Each bit of the DWORD value shows a target service. Turn ON a bit indicating a service supported by the device. (See Table 5-179 for the bit and service correspondence.) Do not write the value with a bit corresponding to an unsupported service turned ON. If not even one service shown in Table 5-179 is supported, omit this or write "0x00000000". If this is omitted, it will be regarded to be "0x00000000". To show that there is a need to support multiple services, write the value with multiple bits turned ON.

(a) Item description example of SupportSafetyCommService element

```
<p:commIfInfoMember label="SupportSafetyCommService">
   <p:label2><p:item>SupportSafetyCommService</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN_Safety</p:item></p:category>
   <p:name><p:item>Support Safety Communication Service</p:item></p:name>
   <p:datatype><p:item>DWORD</p:item></p:datatype>
   <p:data><p:item>0x00000001</p:item></p:data>
</p:commIfInfoMember>
```

Table 5-179 Service Shown by Each Bit

Bit Position	Service
0	IESS-Start
1	IESS-InvokeFunc
2 to 31	For future expansion (Set to 0).

For details on safety communication services, see "CC-Link IE Safety Communication Function Specifications (Application Layer and Service Protocol) BAP-C2007ENG-002 and (Communication Profile) BAP-C2007ENG-003".

(104) Item description of SupportSafetyInvokeFunc element

Table 5-180 Item Description Specifications of SupportSafetyInvokeFunc Element

No.	Item Name	Description Details
1.	LABEL	Write "SupportSafetyInvokeFunc".
2.	LABEL2	Write "SupportSafetyInvokeFunc" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN_Safety".
4.	NAME	Write "Support Safety Function" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "DWORD".
6.	DATA	Each bit of the DWORD value shows a target safety function. Turn ON a bit indicating a safety function supported by the device. (For the correspondence between bit and safety function, See Table 5-181.) Do not write the value with a bit corresponding to an unsupported safety function turned ON. If not even one safety function shown in Table 5-181 is supported, write "0x00000000". If this value is omitted,"Station unique setting information check code verification" (0x00000002) is regarded to be ON. To show that there is a need to support multiple safety function, write the value with multiple bits turned ON. This can be used when the IESS-InvokeFunc service of the SupportSafetyCommService element is ON. (For details on bits of IESS-InvokeFunc, see Table 5-179.)

(a) Item description example of SupportSafetyInvokeFunc element

```
<p:commIfInfoMember label="SupportSafetyInvokeFunc">
   <p:label2><p:item>SupportSafetyInvokeFunc</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN_Safety</p:item></p:category>
   <p:name><p:item>Support Safety Function</p:item></p:name>
   <p:datatype><p:item>DWORD</p:item></p:datatype>
   <p:data><p:item>0x00000002</p:item></p:data>
</p:commIfInfoMember>
```

Table 5-181 Safety Function Shown by Each Bit

Bit Position	Safety Function
0	Station unique ID information verification
1	Station unique setting information check code verification
2	Station unique setting information writing
3 to 31	For future expansion (Set to 0).

For details on safety functions, see "CC-Link IE Safety Communication Function Specifications (Application Layer and Service Protocol) BAP-C2007ENG-002 and (Communication Profile) BAP-C2007ENG-003".

(105) Item description of S_Safe_DefaultSize element

Table 5-182 Item Description Specifications of S_Safe_DefaultSize Element

No.	Item Name	Description Details
1.	LABEL	Write "S_Safe_DefaultSize".
2.	LABEL2	Write "S_Safe_DefaultSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN_Safety".
4.	NAME	Write "Safety Send Bit Data Default Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the default size of the safety send bit data in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum".

(a) Item description example of S_Safe_DefaultSize element

```
<p:commIfInfoMember label="S_Safe_DefaultSize">
   <p:label2><p:item>S_Safe_DefaultSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN_Safety</p:item></p:category>
   <p:name><p:item>Safety Send Bit Data Default Size</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>32</p:item></p:data>
</p:commIfInfoMember>
```

(106) Item description of S_Safe_MaxSize element

Table 5-183 Item Description Specifications of S_Safe_MaxSize Element

No.	Item Name	Description Details
1.	LABEL	Write "S_Safe_MaxSize".
2.	LABEL2	Write "S_Safe_MaxSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN_Safety".
4.	NAME	Write "Safety Send Bit Data Maximum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the maximum size of the safety send bit data in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum".

(a) Item description example of S_Safe_MaxSize element

```
<p:commIfInfoMember label="S_Safe_MaxSize">
   <p:label2><p:item>S_Safe_MaxSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN_Safety</p:item></p:category>
   <p:name><p:item>Safety Send Bit Data Maximum Size</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>256</p:item></p:data>
</p:commIfInfoMember>
```

(107) Item description of S_Safe_MinSize element

Table 5-184 Item Description Specifications of S_Safe_MinSize Element

No.	Item Name	Description Details
1.	LABEL	Write "S_Safe_MinSize".
2.	LABEL2	Write "S_Safe_MinSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN_Safety".
4.	NAME	Write "Safety Send Bit Data Minimum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the minimum size of the safety send bit data in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum".

(a) Item description example of S_Safe_MinSize element

```
<p:commIfInfoMember label="S_Safe_MinSize">
   <p:label2><p:item>S_Safe_MinSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN_Safety</p:item></p:category>
   <p:name><p:item>Safety Send Bit Data Minimum Size</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>2</p:item></p:data>
</p:commIfInfoMember>
```

(108) Item description of S_Safe_Address element

Table 5-185 Item Description Specifications of S_Safe_Address Element

No.	Item Name	Description Details
1.	LABEL	Write "S_Safe_Address".
2.	LABEL2	Write "S_Safe_Address" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN_Safety".
4.	NAME	Write "Safety Send Bit Data Address" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "DWORD".
6.	DATA	Write the start address of the data to be stored in the sub payload for safety send data communication.

(a) Item description example of S_Safe_Address element

```
<p:commIfInfoMember label="S_Safe_Address">
   <p:label2><p:item>S_Safe_Address</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN_Safety</p:item></p:category>
   <p:name><p:item>Safety Send Bit Data Address</p:item></p:name>
   <p:datatype><p:item>DWORD</p:item></p:datatype>
   <p:data><p:item>0x00000123</p:item></p:data>
</p:commIfInfoMember>
```

(109) Item description of R_Safe_DefaultSize element

Table 5-186 Item Description Specifications of R_Safe_DefaultSize Element

No.	Item Name	Description Details
1.	LABEL	Write "R_Safe_DefaultSize".
2.	LABEL2	Write "R_Safe_DefaultSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN_Safety".
4.	NAME	Write "Safety Receive Bit Data Default Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the default size of the safety receive bit data in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum".

(a) Item description example of R_Safe_DefaultSize element

```
<p:commIfInfoMember label="R_Safe_DefaultSize">
   <p:label2><p:item>R_Safe_DefaultSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN_Safety</p:item></p:category>
   <p:name><p:item>Safety Receive Bit Data Default Size</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>16</p:item></p:data>
</p:commIfInfoMember>
```

(110) Item description of R_Safe_MaxSize element

Table 5-187 Item Description Specifications of R_Safe_MaxSize Element

No.	Item Name	Description Details
1.	LABEL	Write "R_Safe_MaxSize".
2.	LABEL2	Write "R_Safe_MaxSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN_Safety".
4.	NAME	Write "Safety Receive Bit Data Maximum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the maximum size of the safety receive bit data in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum".

(a) Item description example of R_Safe_MaxSize element

```
<p:commIfInfoMember label="R_Safe_MaxSize">
   <p:label2><p:item>R_Safe_MaxSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN_Safety</p:item></p:category>
   <p:name><p:item>Safety Receive Bit Data Maximum Size</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>512</p:item></p:data>
</p:commIfInfoMember>
```

(111) Item description of R_Safe_MinSize element

Table 5-188 Item Description Specifications of R_Safe_MinSize Element

No.	Item Name	Description Details
1.	LABEL	Write "R_Safe_MinSize".
2.	LABEL2	Write "R_Safe_MinSize" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN_Safety".
4.	NAME	Write "Safety Receive Bit Data Minimum Size" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "UINT32".
6.	DATA	Write the minimum size of the safety receive bit data in the number of bits. Write a numerical value equal to or larger than 0. The value must be within the range "minimum ≤ default ≤ maximum".

(a) Item description example of R_Safe_MinSize element

```
<p:commIfInfoMember label="R_Safe_MinSize">
   <p:label2><p:item>R_Safe_MinSize</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN_Safety</p:item></p:category>
   <p:name><p:item>Safety Receive Bit Data Minimum Size</p:item></p:name>
   <p:datatype><p:item>UINT32</p:item></p:datatype>
   <p:data><p:item>1</p:item></p:data>
</p:commIfInfoMember>
```

(112) Item description of R_Safe_Address element

Table 5-189 Item Description Specifications of R_Safe_Address Element

No.	Item Name	Description Details
1.	LABEL	Write "R_Safe_Address".
2.	LABEL2	Write "R_Safe_Address" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON_CC-Link_IE_TSN_Safety".
4.	NAME	Write "Safety Receive Bit Data Address" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "DWORD".
6.	DATA	Write the start address of the sub payload for safety receive data communication.

(a) Item description example of R_Safe_Address element

```
<p:commIfInfoMember label="R_Safe_Address">
   <p:label2><p:item>R_Safe_Address</p:item></p:label2>
   <p:category><p:item>COMMON_CC-Link_IE_TSN_Safety</p:item></p:category>
   <p:name><p:item>Safety Receive Bit Data Address</p:item></p:name>
   <p:datatype><p:item>DOWRD</p:item></p:datatype>
   <p:data><p:item>0x00000789</p:item></p:data>
</p:commIfInfoMember>
```

5.3.2 COMM_IF_INPUT part

5.3.2.1 Overview

A COMM_IF_INPUT part describes information related to the input information of the communication interface.

Describe a COMM_IF_INPUT part when there is information to be outputted from the control side of the target module. The part applies to the remote input RX area of the remote I/O module, the A/D conversion completed flag of the digital analog converter, the digital output value of the analog digital converter, etc.

The elements that make up the COMM_IF_INPUT part are defined in accordance with the target module.

The configuration of each element of the COMM_IF_INPUT part, that is, the items that should be described within each element, is the same. Figure 5-16 shows the configuration of an element of the COMM_IF_INPUT part.

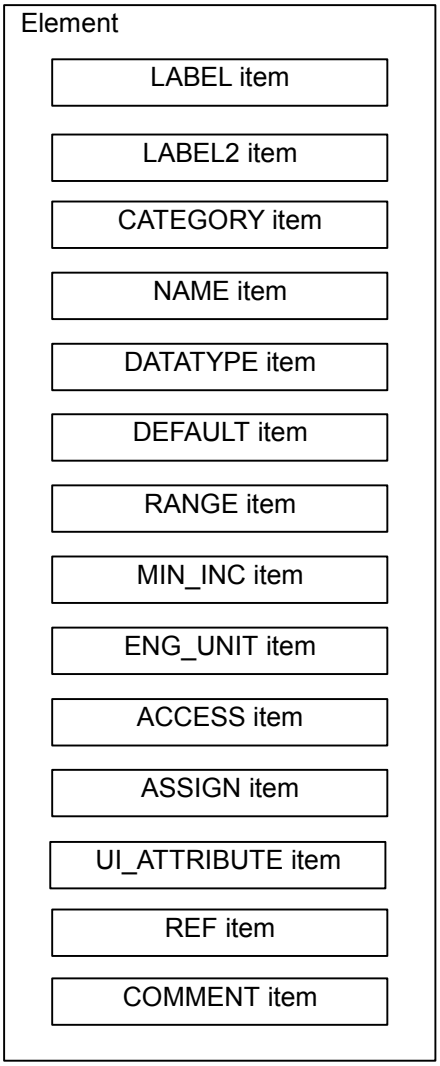


Figure 5-16 Structure Definition of Element of COMM_IF_INPUT and COMM_IF_OUTPUT Part

A single element is described as a single XML element.

Table 5-190 shows the items included in an element of the COMM_IF_INPUT part.

Table 5-190 List of Elements Defined in COMM_IF_Input Part and COMM_IF_OUTPUT Part

No.	Element Name	Description Details	Description Specifications	Required /Optional
1.	LABEL	Describes the label for identifying the element	See Section 4.3.1.19	Required
2.	LABEL2	Describes the label for identifying the element (Used when the utility software supports another language.)	See Section 4.3.1.20	Optional
3.	CATEGORY	Describes the category for grouping the element	See Section 4.3.1.4	Optional
4.	NAME	Describes the name of the element Used when the name and contents of the element are displayed with the utility software.	See Section 4.3.1.24	Optional
5.	DATATYPE	Describes the data type of the element	See Section 4.3.1.10	Optional
6.	DEFAULT	Describes the default value set in the element	See Section 4.3.1.11	Optional
7.	RANGE	Describes the setting range of the element	See Section 4.3.1.26	Optional
8.	MIN_INC	Describes the minimum increment applied to the value of the element	See Section 4.3.1.23	Optional
9.	ENG_UNIT	Describes the engineering unit applied to the value of the element	See Section 4.3.1.13	Optional
10.	ACCESS	Describes the access attribute of the element	See Section 4.3.1.1	Optional
11.	ASSIGN	Describes the remote input/output and remote register assigned to the value of the element	See Section 4.3.1.3	Optional
12.	UI_ATTRIBUTE	Describes the display method when the element is displayed in the utility software	See Section 4.3.1.43	Optional
13.	REF	Describes a reference to the BLOCK_OUTPUT part for a COMM_IF_INPUT part Describes a reference to the BLOCK_INPUT part for a COMM_IF_OUTPUT part	See Section 4.3.1.30	Optional
14.	COMMENT	Describes the meaning of the element as well as usage precautions.	See Section 4.3.1.6	Optional

5.3.2.2 XML format of COMM_IF_INPUT part

In XML format, the COMM_IF_INPUT part is described as an XML element having the XML element name "commIfInput". Table 5-191 shows a list of the attributes of the commIfInput element, and Table 5-192 shows a list of the XML child elements.

Table 5-191 Attribute List for commIfInput Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the COMM_IF_INPUT part. (See Section 4.3.1.19.) (Example: "CommIfInput")	Required

Table 5-192 XML Child Element List for commIfInput Element

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the COMM_IF_INPUT part (See Section 4.3.1.6.)	Optional
2.	commIfInputMember	Describes the contents of the elements of the COMM_IF_INPUT part	Required

In XML format, the element of the COMM_IF_INPUT part is described as an XML element having the XML element name "commIfInputMember".

Table 5-193 shows a list of attributes of the commIfInputMember element, and Table 5-194 shows a list of XML child elements.

Table 5-193 Attribute List for commIfInputMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element. (See Section 4.3.1.19.)	Required

Table 5-194 XML Child Element List of commIfInputMember Element

No.	XML Element Name	Description Details
1.	label2	Describes the LABEL2 item.
2.	category	Describes the CATEGORY item.
3.	name	Describes the NAME item.
4.	datatype	Describes the DATATYPE item.
5.	default	Describes the DEFAULT item.
6.	range	Describes the RANGE item.
7.	minInc	Describes the MIN_INC item.
8.	engUnit	Describes the ENG_UNIT item.
9.	access	Describes the ACCESS item.
10.	assign	Describes the ASSIGN item.
11.	uiAttribute	Describes the UI_ATTRIBUTE item.
12.	ref	Describes the REF item.
13.	comment	Describes the COMMENT item.

Write the COMM_IF_INPUT part using a format such as shown below.

```
<p:commIfInput label=''Write in accordance with the description details of Table 5-191
'Attribute Name label'''>
   <p:comment>''Write in accordance with the description details of Table 5-192 'XML Element
   Name comment'''</p:comment>
   <p:commIfInputMember label=''Write in accordance with the description details of Table
   5-193 'Attribute Name label'''>
      <p:label2>''Write in accordance with the description details of Table 5-194 'XML Element
      Name label2'''</p:label2>
      <p:category>''Write in accordance with the description details of Table 5-194 'XML
      Element Name categor'y''</p:category>
      <p:name>''Write in accordance with the description details of Table 5-194 'XML Element
      Name name'''</p:name>
      <p:datatype>''Write in accordance with the description details of Table 5-194 'XML
      Element Name datatype'''</p:datatype>
      <p:default>''Write in accordance with the description details of Table 5-194'XML
      Element Name default'''</p:default>
      <p:range>''Write in accordance with the description details of Table 5-194 'XML Element
      Name range'''</p:range>
      <p:minInc>''Write in accordance with the description details of Table 5-194 'XML Element
      Name minInc''</p:minInc>
      <p:engUnit>''Write in accordance with the description details of Table 5-194 'XML
      Element Name engUnit'''</p:engUnit>
      <p:access>''Write in accordance with the description details of Table 5-194 'XML Element
      Name access'''</p:access>
      <p:assign>''Write in accordance with the description details of Table 5-194 'XML Element
      Name assign'''</p:assign>
      <p:uiAttribute>''Write in accordance with the description details of Table 5-194 'XML
      Element Name uiAttribute'''</p:uiAttribute>
      <p:ref>''Write in accordance with the description details of Table 5-194 'XML Element
      Name ref'''</p:ref>
      <p:comment>''Write in accordance with the description details of Table 5-194 'XML
      Element Name comment'''</p:comment>
   </p:commIfInputMember>
   <p:commIfInputMember label=''Write in accordance with the description details of Table
   5-193 'Attribute Name label'''>
   :
   :
   </p:commIfInputMember>
   :
   :
</p:commIfInput>
```

5.3.2.3 Description example of each element

The following shows item description examples of the elements of the COMM_IF_INPUT part. In the examples, the element name is set to "CH1_digital_output_val".

Table 5-195 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "CH1_digital_output_val"
2.	LABEL2	Write "CH1_digital_output_val"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "CH1 Digital Output Value"
5.	DATATYPE	Write "INT16"
6.	DEFAULT	In this case, the value of the reference of the REF item is used. Not written.
7.	RANGE	In this case, the value of the reference of the REF item is used. Not written.
8.	MIN_INC	In this case, the value of the reference of the REF item is used. Not written.
9.	ENG_UNIT	In this case, the value of the reference of the REF item is used. Not written.
10.	ACCESS	Write "RF"
11.	ASSIGN	Write "RWr0"
12.	UI_ATTRIBUTE	Omittable item. Not written.
13.	REF	Write "BlockSection.BlockOutput.CH1_digital_output_val".
14.	COMMENT	Omittable item. Not written.

(a) Item description example of element

```
<p:commIfInputMember label="CH1_digital_output_val">
   <p:label2><p:item>CH1_digital_output_val</p:item></p:label2>
   <p:name><p:item>CH1 Digital Output Value</p:item></p:name>
   <p:datatype><p:item>INT16</p:item></p: datatype >
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RWr0</p:item></p:assign>
   <p:ref><p:item>BlockSection.BlockOutput.CH1_digital_output_val</p:item></p:ref>
</p:commIfInputMember>
```

5.3.3 COMM_IF_OUTPUT part

5.3.3.1 Overview

The COMM_IF_OUTPUT part describes information related to the output information of the communication interface. This part is described when there is information to be inputted to the control side of the target module. The part pertains to the remote output RY area of the remote I/O module, and the digital input value of the digital/analog converter, for example.

The configuration of each element of the COMM_IF_OUTPUT part, that is, the items that should be described within each element, is the same as that of the COMM_IF_INPUT part (see Figure 5-16). A single element is described as a single XML element.

For items included in the elements of the COMM_IF_OUTPUT part, see Table 5-190.

5.3.3.2 XML format of COMM_IF_OUTPUT part

In XML format, the COMM_IF_OUTPUT part is described as an XML element having the XML element name "commIfOutput". Table 5-196 shows a list of the attributes of the commIfOutput element, and Table 5-197 shows a list of the XML child elements.

Table 5-196 Attribute List for commIfOutput Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the COMM_IF_OUTPUT part. (See Section 4.3.1.19.) (Example: "commIfOutput")	Required

Table 5-197 XML Child Element List for commIfOutput Element

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the COMM_IF_OUTPUT part. (See Section 4.3.1.6.)	Optional
2.	commIfOutputMember	Describes the contents of the elements of the COMM_IF_OUTPUT part.	Required

In XML format, the element of the COMM_IF_OUTPUT part is described as an XML element having the XML element name "commIfOutputMember".

Table 5-198 shows a list of attributes of the commIfOutputMember element, and Table 5-199shows a list of XML child elements.

Table 5-198 Attribute List for commIfOutputMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	Label	Describes the label for identifying the element. (See Section 4.3.1.19.)	Required

Table 5-199 XML Child Element List of commIfOutputMember Element

No.	XML Element Name	Description Details
1.	label2	Describes the LABEL2 item
2.	category	Describes the CATEGORY item
3.	name	Describes the NAME item
4.	datatype	Describes the DATATYPE item
5.	default	Describes the DEFAULT item
6.	range	Describes the RANGE item
7.	minInc	Describes the MIN_INC item
8.	engUnit	Describes the ENG_UNIT item
9.	access	Describes the ACCESS item
10.	assign	Describes the ASSIGN item
11.	uiAttribute	Describes the UI_ATTRIBUTE item
12.	ref	Describes the REF item
13.	comment	Describes the COMMENT item

Write the COMM_IF_OUTPUT part using a format such as shown below.

```
<p:commIfOutput label="Write in accordance with the description details of Table 5-196
'Attribute Name label'">
   <p:comment>"Write in accordance with the description details of Table 5-197 'XML Element
   Name comment'"</p:comment>
   <p:commIfOutputMember label="Write in accordance with the description details of Table
   5-198 'Attribute Name label'">
      <p:label2>"Write in accordance with the description details of Table 5-199 'XML Element
      Name label2'"</p:label2>
      <p:category>"Write in accordance with the description details of Table 5-199 'XML
      Element Name category'"</p:category>
      <p:name>"Write in accordance with the description details of Table 5-199 'XML Element
      Name name'"</p:name>
      <p:datatype>"Write in accordance with the description details of Table 5-199 'XML
      Element Name datatype'"</p:datatype>
      <p:default>"Write in accordance with the description details of Table 5-199 'XML
      Element Name default'"</p:default>
      <p:range>"Write in accordance with the description details of Table 5-199 'XML Element
      Name range'"</p:range>
      <p:minInc>"Write in accordance with the description details of Table 5-199 'XML Element
      Name minInc'"</p:minInc>
      <p:engUnit>"Write in accordance with the description details of Table 5-199 'XML
      Element Name engUnit'"</p:engUnit>
      <p:access>"Write in accordance with the description details of Table 5-199 'XML Element
      Name access'"</p:access>
      <p:assign>"Write in accordance with the description details of Table 5-199 'XML Element
      Name assign'"</p:assign>
      <p:uiAttribute>"Write in accordance with the description details of Table 5-199 'XML
      Element Name uiAttribute'"</p:uiAttribute>
      <p:ref>"Write in accordance with the description details of Table 5-199 'XML Element
      Name ref'"</p:ref>
      <p:comment>"Write in accordance with the description details of Table 5-199 'XML
      Element Name comment'"</p:comment>
   </p:commIfOutputMember>
   <p:commIfOutputMember label="Write in accordance with the description details of Table
   5-198 'Attribute Name label'">
   :
   :
   </p:commIfOutputMember>
   :
   :
</p:commIfOutput>
```

5.3.3.3 Description example of each element

The following shows item description examples of the elements of the COMM_IF_OUTPUT part. In the examples, the element name is set to "CH1_ave_time_number_of_times".

Table 5-200 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "CH1_ave_time_number_of_times"
2.	LABEL2	Write "CH1_ave_time_number_of_times"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "CH1 Average Time Number"
5.	DATATYPE	Write "INT16"
6.	DEFAULT	In this case, the value of the reference of the REF item is used
7.	RANGE	In this case, the value of the reference of the REF item is used
8.	MIN_INC	In this case, the value of the reference of the REF item is used
9.	ENG_UNIT	In this case, the value of the reference of the REF item is used
10.	ACCESS	Write "RF"
11.	ASSIGN	Write "RWw1"
12.	UI_ATTRIBUTE	Omittable item. Not written.
13.	REF	Write "BlockSection.BlockInput.CH1_ave_time_number_of_times".
14.	COMMENT	Omittable item. Not written.

(a) Item description example of element

```
<p:commIfOutputMember label="CH1_ave_time_number_of_times">
   <p:label2><p:item>CH1_ave_time_number_of_times</p:item></p:label2>
   <p:name><p:item>CH1 Average Time Number</p:item></p:name>
   <p:datatype><p:item>INT16</p:item></p: datatype >
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RWw1</p:item></p:assign>
   <p:ref><p:item>BlockSection.BlockInput.CH1_ave_time_number_of_times</p:item></p:ref
   >
</p:commIfOutputMember>
```

5.3.4 COMM_IF_PARAMETER part

5.3.4.1 Overview

The COMM_IF_PARAMETER part describes information related to the parameters of the target module.

This part pertains to the voltage/current specification of the analog-digital converter and the CH1 average processing specification, for example. Note that information that cannot be set or used as reference via the communication interface, such as values set using the DIP switch, cannot be written.

The elements that make up the COMM_IF_PARAMETER part are defined in accordance with the communication function of the target module.

The configuration of each element of the COMM_IF_PARAMETER part, that is, the items that should be described within each element, is the same. Figure 5-17 shows the configuration of an element of the COMM_IF_PARAMETER part.

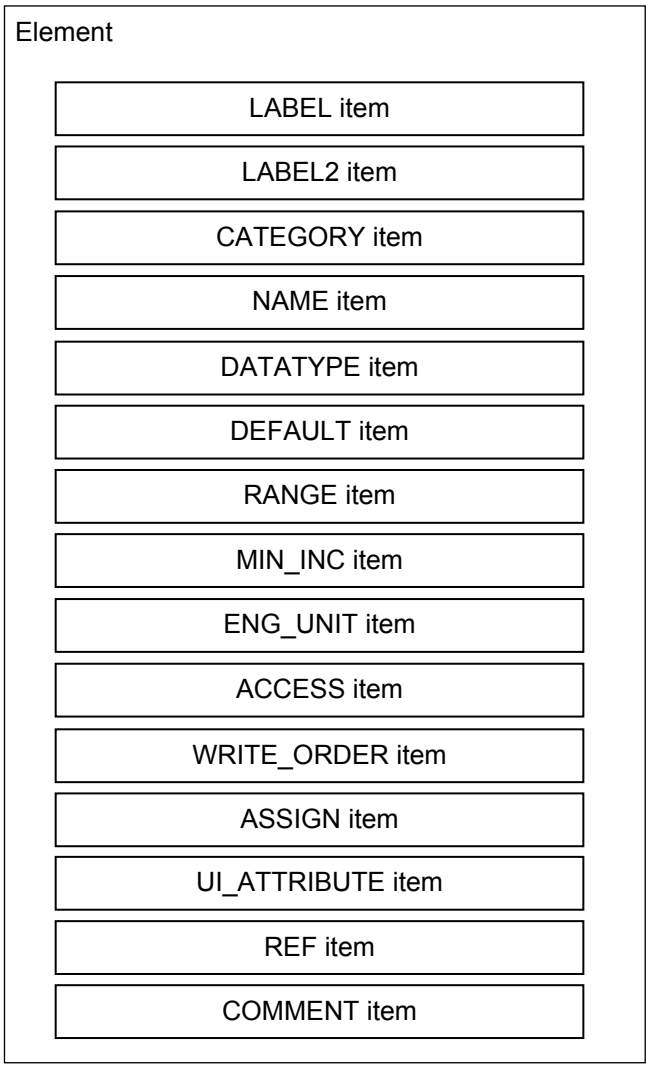


Figure 5-17 Structure Definition for Element of COMM_IF_PARAMETER Part

A single element is described as a single XML element. For element description examples, see Section 5.3.4.3.

The items included in the element are described in Table 5-201.

Table 5-201 Item List for Elements Defined in COMM_IF_PARAMETER Part

No.	Item Name	Description Details	Description Specifications	Required /Optional
1.	LABEL	Describes the label for identifying the element.	See Section 4.3.1.19	Required
2.	LABEL2	Describes the label for identifying the element. (Used when the utility software supports another language.)	See Section 4.3.1.20	Optional
3.	CATEGORY	Describes the category for grouping the element.	See Section 4.3.1.4	Optional
4.	NAME	Describes the name of the element. Used when the name and contents of the element are displayed with the utility software.	See Section 4.3.1.24	Optional
5.	DATATYPE	Describes the data type of the element.	See Section 4.3.1.10	Optional
6.	DEFAULT	Describes the default value set in the element.	See Section 4.3.1.11	Optional
7.	RANGE	Describes the setting range of the element.	See Section 4.3.1.26	Optional
8.	MIN_INC	Describes the minimum increment applied to the value of the element and the ENG_UNIT item.	See Section 4.3.1.23	Optional
9.	ENG_UNIT	Describes the engineering unit applied to the value of the element and the MIN_INC item.	See Section 4.3.1.13	Optional
10.	ACCESS	Describes the access attribute of the element.	See Section 4.3.1.1	Optional
11.	WRITE_ORDER	Describes the order in which elements are to be written to the module.	See Section 4.3.1.46	Optional
12.	ASSIGN	Describes the address and code to be assigned to the element.	See Section 4.3.1.3	*1
13.	UI_ATTRIBUTE	Describes the display method when the element is displayed in the utility software.	See Section 4.3.1.43	Optional
14.	REF	Describes a reference to an element of the BLOCK_PARAMETERreferred to by an element of the communication parameter list.	See Section 4.3.1.30	Optional
15.	COMMENT	Describes the meaning of the element as well as usage precautions.	See Section 4.3.1.6	Optional

*1: This item is required in CC-Link IE TSN and optional in other networks.

5.3.4.2 XML format of COMM_IF_PARAMETER part

In XML format, the COMM_IF_PARAMETER part is described as an XML element having the XML element name "commIfParameter". Table 5-202 shows a list of the attributes of the commIfParameter element, and Table 5-203 shows a list of the XML child elements.

Table 5-202 Attribute List for commIfParameter Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the COMM_IF_PARAMETER part. (See Section 4.3.1.19.)	Required

Table 5-203 XMLChild Element List for commIfParameterElement

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the COMM_IF_PARAMETER part. (See Section 4.3.1.6.)	Optional
2.	commIfParameter Member	Describes the contents of the elements of the COMM_IF_PARAMETER part.	Required

In XML format, the element of the COMM_IF_PARAMETER part is described as an XML element having the XML element name "commIfParameterMember".

Table 5-204 shows a list of attributes of the commIfParameterMember element, and Table 5-205 shows a list of XML child elements.

Table 5-204 Attribute List for commIfParameterMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element. (See Section 4.3.1.19.)	Required

Table 5-205 XML Child Element List of commIfParameterMember Element

No.	XML Element Name	Description Details
1.	label2	Describes the LABEL2 item
2.	category	Describes the CATEGORY item
3.	name	Describes the NAME item
4.	datatype	Describes the DATATYPE item
5.	default	Describes the DEFAULT item
6.	range	Describes the RANGE item
7.	minInc	Describes the MIN_INC item
8.	engUnit	Describes the ENG_UNIT item
9.	access	Describes the ACCESS item
10.	writeOrder	Describes the WRITE_ORDER item
11.	assign	Describes the ASSIGN item
12.	uiAttribute	Describes the UI_ATTRIBUTE item
13.	ref	Describes the REF item
14.	comment	Describes the COMMENT item

Write the COMM_IF_PARAMETER part using a format such as shown below.

```
<p:commIfParameter label="Write in accordance with the description details of Table 5-202
'Attribute Name label'">
   <p:comment>"Write in accordance with the description details of Table 5-203 'XML Element
   Name comment'"</p:comment>
   <p:commIfParameterMember label="Write in accordance with the description details of Table
   5-204 'Attribute Name label'">
      <p:label2>"Write in accordance with the description details of Table 5-205 'XML Element
      Name label2'"</p:label2>
      <p:category>"Write in accordance with the description details of Table 5-205 'XML
      Element Name category'"</p:category>
      <p:name>"Write in accordance with the description details of Table 5-205 'XML Element
      Name name'"</p:name>
      <p:datatype>"Write in accordance with the description details of Table 5-205 'XML
      Element Name datatype'"</p:datatype>
      <p:default>"Write in accordance with the description details of Table 5-205 'XML
      Element Name default'"</p:default>
      <p:range>"Write in accordance with the description details of Table 5-205 'XML Element
      Name range'"</p:range>
      <p:minInc>"Write in accordance with the description details of Table 5-205 'XML Element
      Name minInc'"</p:minInc>
      <p:engUnit>"Write in accordance with the description details of Table 5-205 'XML
      Element Name engUnit'"</p:engUnit>
      <p:access>"Write in accordance with the description details of Table 5-205 'XML Element
      Name access'"</p:access>
      <p:writeOrder>"Write in accordance with the description details of Table 5-205 'XML
      Element Name writeOrder"</p:writeOrder>
      <p:assign>"Write in accordance with the description details of Table 5-205 'XML Element
      Name assign'"</p:assign>
      <p:uiAttribute>"Write in accordance with the description details of Table 5-205 'XML
      Element Name uiAttribute'"</p:uiAttribute>
      <p:ref>"Write in accordance with the description details of Table 5-205 'XML Element
      Name ref'"</p:ref>
      <p:comment>"Write in accordance with the description details of Table 5-205 'XML
      Element Name comment'"</p:comment>
   </p:commIfParameterMember>
   <p:commIfParameterMember label="Write in accordance with the description details of Table
   5-202 'Attribute Name label'">
   :
   :
   </p:commIfParameterMember>
   :
   :
</p:commIfParameter>
```

5.3.4.3 Description example of element

The following shows item description examples of the elements of the COMM_IF_PARAMETER part. In the examples, the element name is set to "Parameter1".

Table 5-206 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "Parameter1"
2.	LABEL2	Write "Parameter1"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "Parameter 1"
5.	DATATYPE	In this case, the value of the reference of the REF item is used. Not written.
6.	DEFAULT	In this case, the value of the reference of the REF item is used. Not written.
7.	RANGE	In this case, the value of the reference of the REF item is used. Not written.
8.	MIN_INC	In this case, the value of the reference of the REF item is used. Not written.
9.	ENG_UNIT	In this case, the value of the reference of the REF item is used. Not written.
10.	ACCESS	In this case, the value of the reference of the REF item is used. Not written.
11.	WRITE_ORDER	In this case, the value of the reference of the REF item is used. Not written.
12.	ASSIGN	Write "<0x5E><0xDE><0x01>" (with XML file: "<0x5E><0xDE><0x01>").
13.	UI_ATTRIBUTE	Omittable item. Not written.
14.	REF	Write "BlockSec.BlockParameter.Parameter1"
15.	COMMENT	Omittable item. Not written.

(a) Item description example of element

```
<p:commIfParameterMember label="Parameter1">
   <p:label2><p:item>Parameter1</p:item></p:label2>
   <p:name><p:item>Parameter 1</p:item></p:name>
   <p:assign><p:item>&lt;0x5E&gt;&lt;0xDE&gt;&lt;0x1&gt;</p:item></p:assign>
   <p:ref><p:item>BlockSec.BlockParameter.Parameter1</p:item></p:ref>
</p:commIfParameterMember>
```

5.3.5 COMM_IF_COMMAND part

5.3.5.1 Overview

The COMM_IF_COMMAND part describes information related to the commands issued by the communication interface. This part pertains to the CH1 conversion enable/disable specification of the analog-digital converter, for example.

The elements that make up the COMM_IF_COMMAND part are defined in accordance with the communication function of the target module.

The configuration of each element of the COMM_IF_COMMAND part, that is, the items that should be described within each element, is the same. Figure 5-18 shows the configuration of an element of the COMM_IF_COMMAND part.

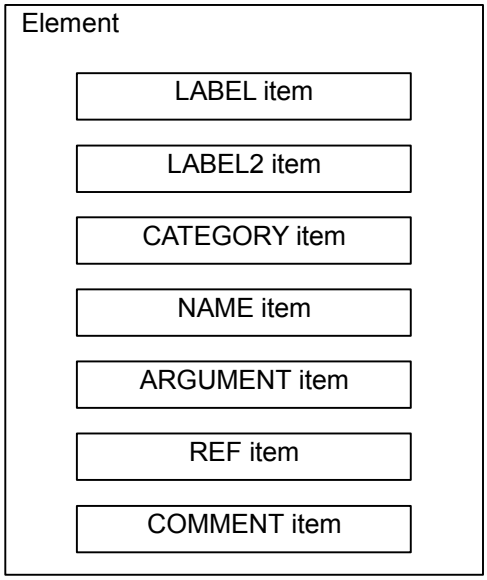


Figure 5-18 Structure Definition for Element of COMM_IF_COMMAND Part

A single element is described as a single XML element. For description examples of each element, see Section 5.3.5.3.

The items included in the element are described in Table 5-207.

Table 5-207 Item List for Elements Defined in COMM_IF_COMMAND Part

No.	Item Name	Description Details	Description Specifications	Required /Optional
1.	LABEL	Describes the label for identifying the element.	See Section 4.3.1.19	Required
2.	LABEL2	Describes the label for identifying the element. (Used when the utility software supports another language.)	See Section 4.3.1.20	Optional
3.	CATEGORY	Describes the category for grouping the element.	See Section 4.3.1.4	Optional
4.	NAME	Describes the name of the item. Used when the name and contents are displayed with the utility software.	See Section 4.3.1.24	Optional
5.	ARGUMENT	Describes the label corresponding to the COMMAND_ARGUMENT part for indicating arguments used by the element.	See Section 4.3.1.3	Optional
6.	REF	Describes the reference to the BLOCK_COMMAND part from the element.	See Section 4.3.1.30	Optional
7.	COMMENT	Describes the meaning of the element as well as usage precautions.	See Section 4.3.1.6	Optional

5.3.5.2 XML format of COMM_IF_COMMAND part

In XML format, the COMM_IF_COMMAND part is described as an XML element having the XML element name "commIfCommand". Table 5-208 shows a list of the attributes of the commIfCommand element, and Table 5-209 shows a list of the XML child elements.

Table 5-208 Attribute List for commIfCommand Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the COMM_IF_COMMAND part. (See Section 4.3.1.19.)	Required

Table 5-209 XML Child Element List for commIfCommandElement

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the COMM_IF_COMMAND part. (See Section 4.3.1.6.)	Optional
2.	commIfCommand Member	Describes the contents of the elements of the COMM_IF_COMMAND part.	Required

In XML format, the element of the COMM_IF_COMMAND part is described as an XML element having the XML element name "commIfCommandMember". Table 5-210 shows a list of attributes of the commIfCommandMember element, and Table 5-211 shows a list of XML child elements.

Table 5-210 Attribute List for commIfPCommandMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element. (See Section 4.3.1.19.)	Required

Table 5-211 XML Child Element List of commIfCommandMember Element

No.	XML Element Name	Description Details
1.	label2	Describes the LABEL2 item
2.	category	Describes the CATEGORY item
3.	name	Describes the NAME item
4.	argument	Describes the ARGUMENT item
5.	ref	Describes the REF item
6.	comment	Describes the COMMENT item

Write the COMM_IF_COMMAND part using a format such as shown below.

```
<p:commIfCommand label="Write in accordance with the description details of Table 5-208
'Attribute Name label'">
   <p:comment>"Write in accordance with the description details of Table 5-209 'XML Element
   Name comment'"</p:comment>
   <p:commIfCommandMember label="Write in accordance with the description details of Table
   5-210 'Attribute Name label'">
      <p:label2>"Write in accordance with the description details of Table 5-211 'XML Element
      Name label2'"</p:label2>
      <p:category>"Write in accordance with the description details of Table 5-211 'XML
      Element Name category'"</p:category>
      <p:name>"Write in accordance with the description details of Table 5-211 'XML Element
      Name name'"</p:name>
      <p:argument>"Write in accordance with the description details of Table 5-211 'XML
      Element Name argument"</p:argument>
      <p:ref>"Write in accordance with the description details of Table 5-211 'XML Element
      Name ref'"</p:ref>
      <p:comment>"Write in accordance with the description details of Table 5-211 'XML
      Element Name comment'"</p:comment>
   </p:commIfCommandMember>
   <p:commIfCommandMember label="Write in accordance with the description details of Table
   5-210 'Attribute Name label'">
   :
   :
   </p:commIfCommandMember>
   :
   :
</p:commIfCommand>
```

5.3.5.3 Description example of element

The following shows an item description example of the elements of the COMM_IF_COMMAND part. In the example, the element name is set to "CmdGetFaultRecord".

Table 5-212 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "CmdGetFaultRecord"
2.	LABEL2	Write "CmdGetFaultRecord"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "Fault Record Acquisition"
5.	ARGUMENT	In this case, the value of the reference of the REF item is used
6.	REF	Write "BlockSection.BlockCommand.CmdGetFaultRecord"
7.	COMMENT	Omittable item. Not written.

(a) Item description example of element

```
<p:commIfCommandMember label="CmdGetFaultRecord">
   <p:label2><p:item>CmdGetFaultRecord</p:item></p:label2>
   <p:name><p:item>Fault Record Acquisition</p:item></p:name>
   <p:ref><p:item>BlockSection.BlockCommand.CmdGetFaultRecord</p:item></p:ref>
</p:commIfCommandMember>
```

5.3.6 METHOD part

5.3.6.1 Overview

The METHOD part provides information related to the communication service.

This part describes the procedures of the I/O read service for the remote input RX and remote register RWr areas, the I/O write service for the remote output RY and remote register RWw areas, the parameter write service for the parameter area, the parameter read service, and the command execution service.

The configuration of each element of the METHOD part, that is, the items that should be described within each element, is the same. Figure 5-19 shows the configuration of an element of the METHOD part.

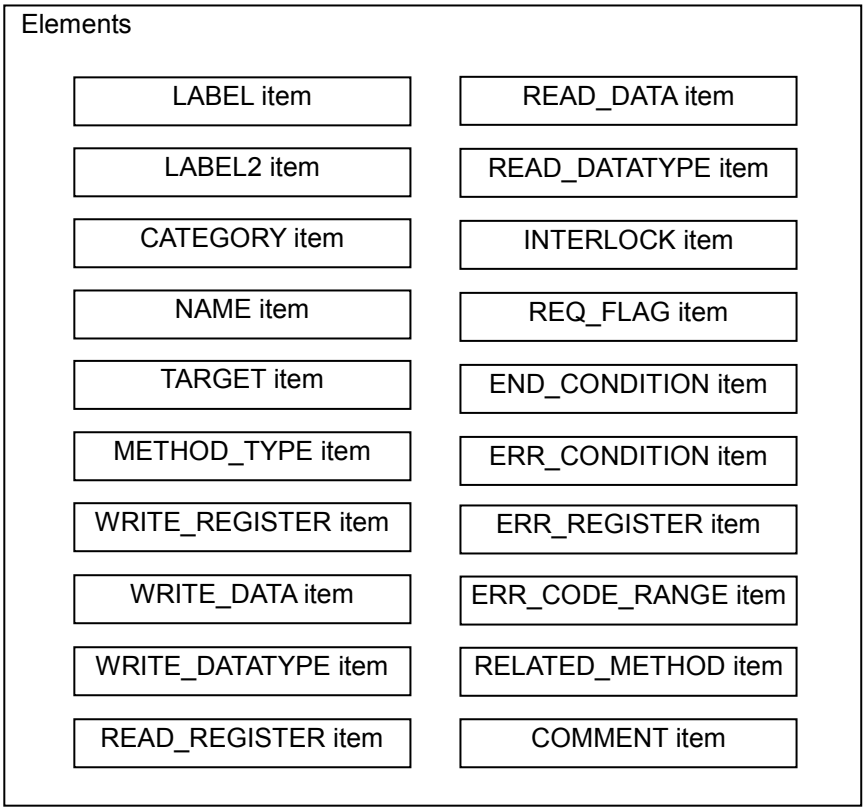


Figure 5-19 Structure Definition for Element of METHOD Part

A single element is described as a single XML element. For details of element description specifications, see Section 5.3.6.4.

The items included in the element are described in Table 5-213.

Table 5-213 Item List for Elements Defined in METHOD Part

No.	Item Name	Description Details	Description Specifications	Required /Optional
1.	LABEL	Describes the label for identifying the element.	See Section 4.3.1.19	Required
2.	LABEL2	Describes the label for identifying the element. (Used when the utility software supports another language.)	See Section 4.3.1.20	Optional
3.	CATEGORY	Describes the category for grouping the element.	See Section 4.3.1.4	Optional
4.	NAME	Describes the name of the element. Used when the name and contents are displayed with the utility software.	See Section 4.3.1.24	Required
5.	TARGET	Describes the element to be processed by the METHOD part based on the same description specifications as the REF item (see Section 4.3.1.30).	See Section 4.3.1.35	Required
6.	METHOD_TYPE	Describes the type of the METHOD part.	See Section 4.3.1.22	Required
7.	WRITE_REGISTER	Describes the remote output and remote register for writing.	See Section 4.3.1.47	Optional
8.	WRITE_DATA	Describes the write value. Describes the same number of write values as in the WRITE_REGISTER item above when specified in multiple in the WRITE_REGISTER item.	See Section 4.3.1.44	Optional
9.	WRITE_DATATYPE	Describes the remote output and remote register data type for writing. Describes the same number of data types as in the WRITE_REGISTER item above when specified in multiple in the WRITE_REGISTER item.	See Section 4.3.1.45	Optional
10.	READ_REGISTER	Describes the remote input and remote register for reading.	See Section 4.3.1.29	Optional
11.	READ_DATA	Describes the reference to the storage destination of the read value. Describes the same number of read value storage destinations as in the READ_REGISTER item when specified in multiple in the READ_REGISTER item.	See Section 4.3.1.27	Optional
12.	READ_DATATYPE	Describes the remote input and remote register data type for reading. Describes the same number of data types as in the READ_REGISTER item when specified in multiple in the READ_REGISTER item.	See Section 4.3.1.28	Optional
13.	INTERLOCK	Describes the interlock remote input/output, remote register, and on/off status or value.	See Section 4.3.1.18	Optional
14.	REQ_FLAG	Describes the request flag for handshake implementation using an assignment expression.	See Section 4.3.1.34	Optional
15.	END_CONDITION	Describes the normal completion conditions for handshake implementation.	See Section 4.3.1.12	Optional
16.	ERR_CONDITION	Describes the abnormal completion conditions for handshake implementation.	See Section 4.3.1.14	Optional
17.	ERR_REGISTER	Describes the remote register that stores values when an error occurs.	See Section 4.3.1.14	Optional
18.	ERR_CODE_RANGE	Describes the error code range.	See Section 4.3.1.15	Optional
19.	RELATED_METHOD	Describes the reference to the METHOD part that indicates the pre-processing of the METHOD part.	See Section 4.3.1.33	Optional
20.	COMMENT	Describes the meaning of the element as well as usage precautions.	See Section 4.3.1.6	Optional

5.3.6.2 XML description of METHOD part

In XML format, the METHOD part is described as an XML element having the XML element name "method". Table 5-214 shows a list of the attributes of the method element, and Table 5-215 shows a list of the XML child elements.

Table 5-214 Attribute List for method Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the METHOD part. (See Section 4.3.1.19.)	Required

Table 5-215 XML Child Element List for methodElement

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the METHOD part. (See Section 4.3.1.6.)	Optional
2.	methodMember	Describes the contents of the elements of the METHOD part.	Required

In XML format, the element of the METHOD part is described as an XML element having the XML element name "methodMember". Table 5-216 shows a list of attributes of the methodMember element, and Table 5-217 shows a list of XML child elements.

Table 5-216 Attribute List for methodMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element. (See Section 4.3.1.17.)	Required

Table 5-217 XML Child Element List of methodMember Element

No.	XML Element Name	Description Details	Required/Optional
1.	label2	Describes the LABEL2 item	Optional
2.	category	Describes the CATEGORY item	Optional
3.	name	Describes the NAME item	Required
4.	target	Describes the item	Required
5.	methodType	Describes the METHOD_TYPE item	Required
6.	writeRegister	Describes the WRITE_REGISTER item	Optional
7.	writeData	Describes the WRITE_DATA item	Optional
8.	writeDatatype	Describes the WRITE_DATATYPE item	Optional
9.	readRegister	Describes the READ_REGISTER item	Optional
10.	readData	Describes the READ_DATA item	Optional
11.	readDatatype	Describes the READ_DATATYPE item	Optional
12.	interlock	Describes the INTERLOCK item	Optional
13.	reqFlag	Describes the REQ_FLAG item	Optional
14.	endCondition	Describes the END_CONDITION item	Optional
15.	errCondition	Describes the ERR_CONDITION item	Optional
16.	errRegister	Describes the ERR_REGISTER item	Optional
17.	errCodeRange	Describes the ERR_CODE_RANGE item	Optional
18.	relatedMethod	Describes the RELATED_METHOD item	Optional
19.	comment	Describes the COMMENT item	Optional

Write the METHOD part using a format such as shown below.

```
<p:method label="Write in accordance with the description details of Table 5-214 'Attribute
Name label'">
   <p:comment>"Write in accordance with the description details of Table 5-215 'XML Element
   Name comment'"</p:comment>
   <p:methodMember label="Write in accordance with the description details of Table 5-216
   'Attribute Name label'">
      <p:label2>"Write in accordance with the description details of Table 5-217 'XML Element
      Name label2'"</p:label2>
      <p:category>"Write in accordance with the description details of Table 5-217 'XML
      Element Name category'"</p:category>
      <p:name>"Write in accordance with the description details of Table 5-217 'XML Element
      Name name'"</p:name>
      <p:target>"Write in accordance with the description details of Table 5-217 'XML Element
      Name target'"</p:target>
      <p:methodType>"Write in accordance with the description details of Table 5-217 'XML
      Element Name methodType'"</p:methodType>
      <p:writeRegister>"Write in accordance with the description details of Table 5-217 'XML
      Element Name writeRegister'"
</p:writeRegister>
      <p:writeData>"Write in accordance with the description details of Table 5-217 'XML
      Element Name writeData'"</p:writeData>
      <p:writeDatatype>"Write in accordance with the description details of Table 5-217 'XML
      Element Name writeDatatype'"
</p:writeDatatype>
      <p:readRegister>"Write in accordance with the description details of Table 5-217 'XML
      Element Name readRegister'"</p:readRegister>
      <p:readData>"Write in accordance with the description details of Table 5-217 'XML
      Element Name readData'"</p:readData>
      <p:readDatatype>"Write in accordance with the description details of Table 5-217 'XML
      Element Name readDatatype'"</p:readDatatype>
      <p:interlock>"Write in accordance with the description details of Table 5-217 'XML
      Element Name interlock'"</p:interlock>
      <p:reqFlag>"Write in accordance with the description details of Table 5-217 'XML
      Element Name reqFlag'"</p:reqFlag>
      <p:endCondition>"Write in accordance with the description details of Table 5-217 'XML
      Element Name endCondition'"</p:endCondition>
      <p:errCondition>"Write in accordance with the description details of Table 5-217 'XML
      Element Name errCondition'"</p:errCondition>
      <p:errRegister>"Write in accordance with the description details of Table 5-217 'XML
      Element Name errRegister'"</p:errRegister>
      <p:errCodeRange>"Write in accordance with the description details of Table 5-217 'XML
      Element Name errCodeRange'"</p:errCodeRange>
      <p:relatedMethod>"Write in accordance with the description details of Table 5-217 'XML
      Element Name relatedMethod'"
</p:relatedMethod>
      <p:comment>"Write in accordance with the description details of Table 5-217 'XML
      Element Name comment'"</p:comment>
   </p:methodMember>
   <p:methodMember label="Write in accordance with the description details of Table 5-216
   'Attribute Name label'">
      :
      :
   </p:methodMember>
      :
      :
</p:method>
```

5.3.6.3 METHOD operation

When a procedure-based communication service is executed by the utility software on a module, a procedure such as described below is performed (see Figure 5-20).

(1) Verify interlock

(2) Write setting value

(3) Set request flag

(4) Verify completion

(5) Read acquired value

(6) Clear request

This procedure is developed into a model as METHOD, describing remote register and shared memory based procedures as well as data area information.

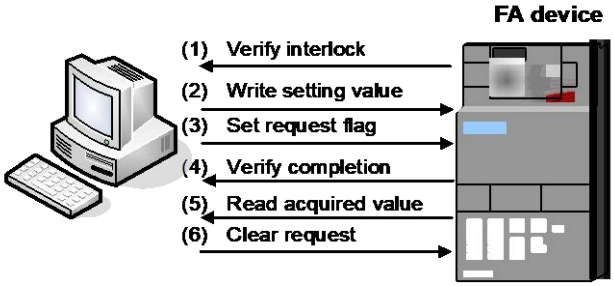


Figure 5-20 Flow of Procedure-Based Communication Service

The following items are described in METHOD. For details of these items, see Table 5-213 of Section 5.3.6.1.

- TARGET
- WRITE_DATA
- READ_DATA
- WRITE_REGISTER
- READ_REGISTER
- INTERLOCK
- REQ_FLAG
- ERR_REGISTER
- END_CONDITION
- ERR_CONDITION

Figure 5-21 shows an operation image of a communication service based on the above items.

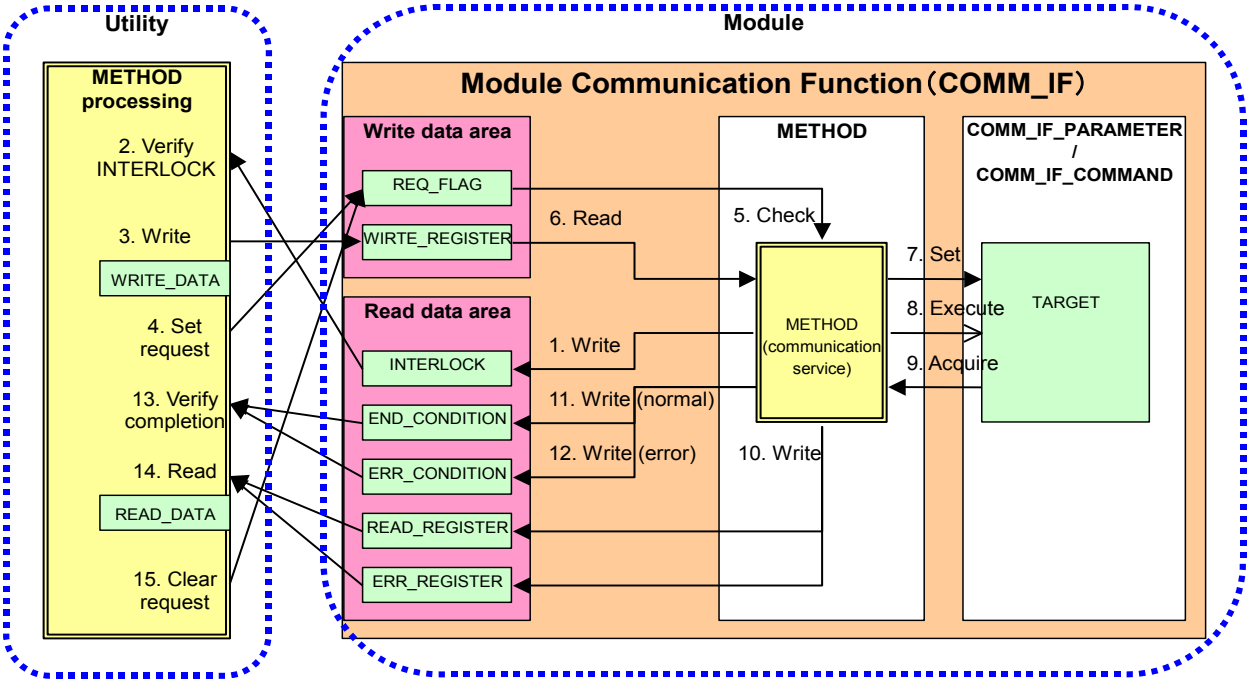


Figure 5-21 Operation of Communication Service Described in METHOD

The following shows the METHOD call-up and operation sequence. Figure 5-22 shows a sequence diagram of the sequence.

1. Remove INTERLOCK conditions when the communication service described in METHOD becomes executable.
2. Read the data area written in INTERLOCK and assess whether or not the communication service described in METHOD is executable.
3. Write the command codes and setting values described in WRITE_DATA to WRITE_REGISTER.
4. Set the REQ_FLAG for requesting the communication service.
5. Check if the REQ_FLAG is set and, if so, execute the communication service of Step 6 and beyond.
6. Read the command codes and setting values from WRITE_REGISTER.
7. Set the setting values in TARGET in accordance with the communication service and command codes.
8. Execute the processing of TARGET in accordance with the communication service and command codes.
9. Get the acquired values of TARGET in accordance with the communication service and command codes.
10. Write the acquired values, statuses, and error codes to READ_REGISTER and ERR_REGISTER.
11. Set END_CONDITION when the communication service ends normally.
12. Set ERR_CONDITION when the communication service ends abnormally.
13. Monitor the END_CONDITION and ERR_CONDITION, and verify completion.
14. Once completion is verified, read the acquired values, statuses, and error codes of READ_REGISTER and ERR_REGISTER.
15. Clear the REQ_FLAG.
16. Check if the REQ_FLAG has been cleared and, if so, clear END_CONDITION and ERR_CONDITION. (Note: Omitted in Figure 5-22 due to illustration complexities.)

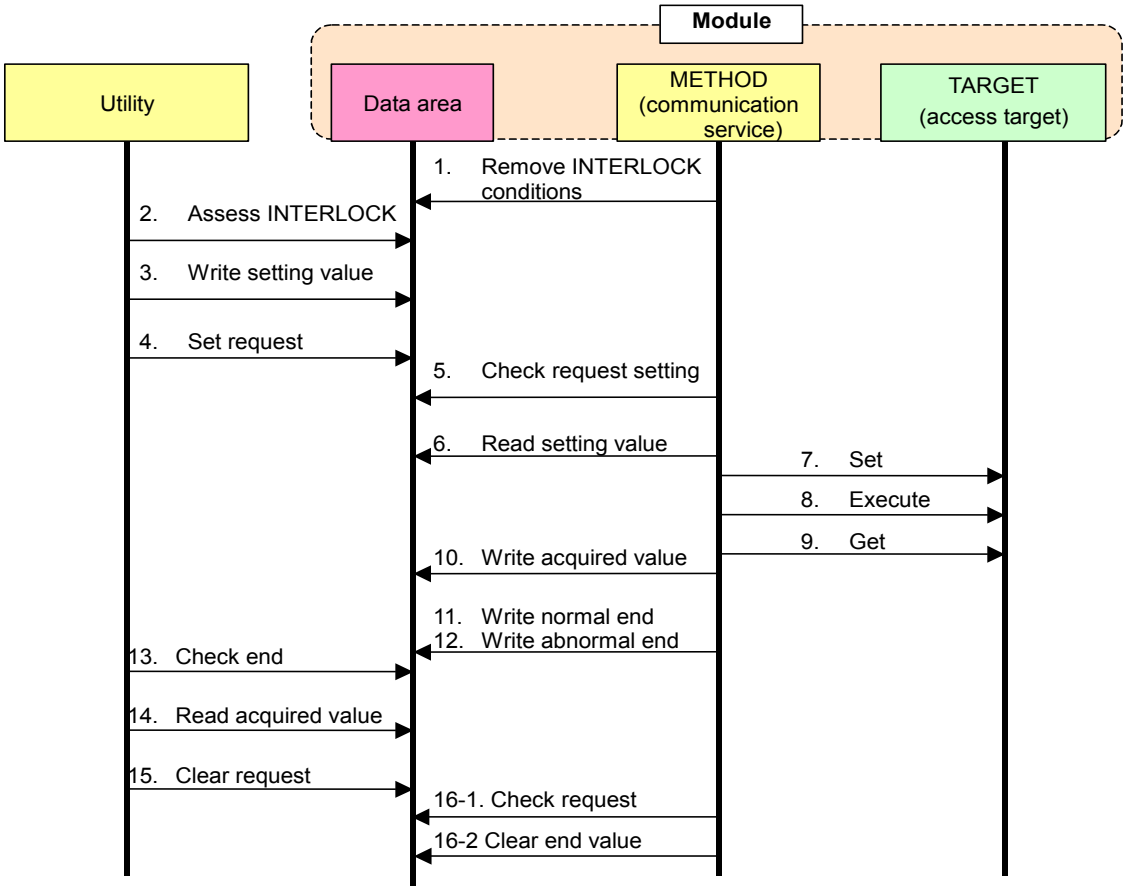


Figure 5-22 METHOD Sequence

5.3.6.4 Description example of element

The following shows an item description example of an element of the METHOD part. In the example below, the element name is set to "Method1".

Table 5-218 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "Method1"
2.	LABEL2	Write "Method1"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "Read Parameter"
5.	TARGET	Write "CommParameter"
6.	METHOD_TYPE	Write "PARAMETER"
7.	WRITE_REGISTER	Write "RWw9"
8.	WRITE_DATA	Write "$(ASSIGN)"
9.	WRITE_DATATYPE	Write "UINT16"
10.	READ_REGISTER	Write "RWr9"
11.	READ_DATA	Write "$(VALUE)"
12.	READ_DATATYPE	Write "$(DATATYPE)"
13.	INTERLOCK	Write "RX1B==ON"
14.	REQ_FLAG	Write "RY0F==ON"
15.	END_CONDITION	Write "RX0F==ON"
16.	ERR_CONDITION	Write "RX1A==ON"
17.	ERR_REGISTER	Omittable item. Not written.
18.	ERR_CODE_RANGE	Omittable item. Not written.
19.	RELATED_METHOD	Omittable item. Not written.
20.	COMMENT	Omittable item. Not written.

(a) Item description example of element

```
<p:methodMember label="Method1">
   <p:label2><p:item>Method1</p:item></p:label2>
   <p:name><p:item>Read Parameter</p:item></p:name>
   <p:target><p:item>CommParameter.*</p:item></p:target>
   <p:methodType><p:item>PARAMETER</p:item></p:methodType>
   <p:writeRegister><p:item>RWw9</p:item></p:writeRegister>
   <p:writeData><p:item>$(ASSIGN)</p:item></p:writeData>
   <p:writeDatatype><p:item>UINT16</p:item></p:writeDatatype>
   <p:readRegister><p:item>RWr9</p:item></p:readRegister>
   <p:readData><p:item>$(VALUE)</p:item></p:readData>
   <p:readDatatype><p:item>$(DATATYPE)</p:item></p:readDatatype>
   <p:interlock><p:item>RX1B==ON</p:item></p:interlock>
   <p:reqFlag><p:item>RY0F=ON</p:item></p:reqFlag>
   <p:endCondition><p:item>RX0F==ON</p:item></p:endCondition>
   <p:errCondition><p:item>RX1A==ON</p:item></p:errCondition>
</p:methodMember>
```

5.3.7 MESSAGE part

5.3.7.1 Overview

The MESSAGE part, similar to the METHOD part, describes information related to the commands to be issued from the communication interface and the parameter setup procedure. The MESSAGE part describes the commands that use transient commands, and the data format for parameter setup.

The elements that make up the MESSAGE part are defined in accordance with the communication function of the target module.

Figure 5-23 shows the configuration of an element of the MESSAGE part. The items described in the element differ for each part.

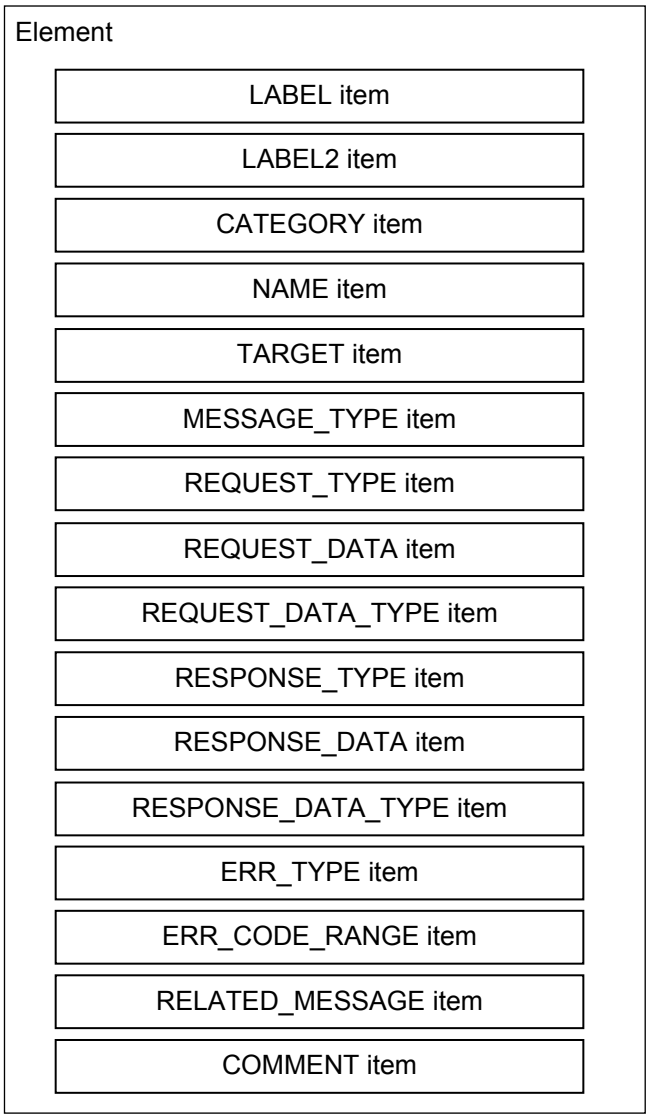


Figure 5-23 Structure Definition for Element of MESSAGE Part

A single element is described as a single XML element. For details of the description specifications of each element, see Section 5.3.7.4.

The following describes the procedure information required for parameter setup and command execution with the items of the MESSAGE part.

Table 5-219 Item List for Elements Defined in MESSAGE Part

No.	Item Name	Description Details	Description Specifications	Required /Optional
1.	LABEL	Describes the label for identifying the element. Write "SLMP" as a prefix for an SLMP-based MESSAGE. Example: SLMPGetParam, SLMPInvReset	See Section 4.3.1.19	Required
2.	LABEL2	Describes the label for identifying the element. (Used when the utility software supports another language.)	See Section 4.3.1.20	Optional
3.	CATEGORY	Describes the category for grouping the element.	See Section 4.3.1.4	Optional
4.	NAME	Describes the name of the element. Used when the name and contents are displayed with the utility software.	See Section 4.3.1.24	Required
5.	TARGET	Describes the element to be processed in the MESSAGE part.	See Section 4.3.1.35	Required
6.	MESSAGE_TYPE	Describes the type of the MESSAGE part.	See Section 4.3.1.21	Required
7.	REQUEST_TYPE	Describes the data format type for executing request processing.	See Section 4.3.1.37	Required*1
8.	REQUEST_DATA	Describes the value for request processing.	See Section 4.3.1.35	Optional
9.	REQUEST_DATA_TYPE	Describes the data type of the REQUEST_DATA item.	See Section 4.3.1.36	Optional
10.	RESPONSE_TYPE	Describes the type of the data format used in response processing.	See Section 4.3.1.40	Optional
11.	RESPONSE_DATA	Describes the value to be returned by response processing.	See Section 0	Optional
12.	RESPONSE_DATA_TYPE	Describes the data type of the RESPONSE_DATA item.	See Section 4.3.1.39	Optional
13.	ERR_TYPE	Describes the type of the data format used in the response processing when an error occurs.	See Section 4.3.1.17	Optional
14.	ERR_CODE_RANGE	Describes the error code range.	See Section 4.3.1.15	Optional
15.	RELATED_MESSAGE	Describes the reference to a MESSAGE part that indicates the pre-processing of the MESSAGE part.	See Section 4.3.1.32	Optional
16.	COMMENT	Describes the meaning of the element as well as usage precautions.	See Section 4.3.1.6	Optional

*1. It may not be required for CC-Link IE TSN. For details, see REQUEST_TYPE conventions (Section 4.3.1.37).

5.3.7.2 XML description of MESSAGE part

In XML format, the MESSAGE part is described as an XML element having the XML element name "message". Table 5-220 shows a list of attributes of the message element, and Table 5-221 shows a list of XML child elements.

Table 5-220 Attribute List for message Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the MESSAGE part.(See Section 4.3.1.19.)	Required

Table 5-221 XML Child Element List of message Element

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comments related to the MESSAGE part. (See Section 4.3.1.6.)	Optional
2.	messageMember	Describes the contents of the element of the MESSAGE part	Required

In XML format, the MESSAGE part is described as an XML element having the XML element name "messageMember". Table 5-222 shows a list of the attributes of the messageMember element, and Table 5-223 shows a list of the XML child elements.

Table 5-222 Attribute List for messageMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element. (See Section 4.3.1.19.)	Required

Table 5-223 XML Child Element List of messageMember Element

No.	XML Element Name	Description Details
1.	label2	Describes the LABEL2 item.
2.	category	Describes the CATEGORY item.
3.	name	Describes the NAME item.
4.	target	Describes the TARGET item.
5.	messageType	Describes the MESSAGE_TYPE item.
6.	requestType	Describes the REQUEST_TYPE item
7.	requestData	Describes the REQUEST_DATA item
8.	requestDatatype	Describes the REQUEST_DATA_TYPE item
9.	responseType	Describes the RESPONSE_TYPE item
10.	responseData	Describes the RESPONSE_DATA item
11.	responseDatatype	Describes the RESPONSE_DATA_TYPE item
12.	errType	Describes the ERR_TYPE item
13.	errCodeRange	Describes the ERR_CODE_RANGE item
14.	relatedMessage	Describes the RELATED_MESSAGE item
15.	comment	Describes the COMMENT item

Write the MESSAGE part using a format such as shown below.

```
<p:message label="Write in accordance with the description details of Table 5-220 'Attribute
Name label'">
   <p:comment>"Write in accordance with the description details of Table 5-221 'XML Element
   Name comment'"</p:comment>
   <p:messageMember label="Write in accordance with the description details of Table 5-222
   'Attribute Name label'">
      <p:label2>"Write in accordance with the description details of Table 5-223 'XML Element
      Name label2'"</p:label2>
      <p:category>"Write in accordance with the description details of Table 5-223 'XML
      Element Name category'"</p:category>
      <p:name>"Write in accordance with the description details of Table 5-223 'XML Element
      Name name'"</p:name>
      <p:target>"Write in accordance with the description details of Table 5-223 'XML Element
      Name target'"</p:target>
      <p:messageType>
      "Write in accordance with the description details of Table 5-223 'XML Element Name
      messageType'"
      </p:messageType>
      <p:requestType>
      "Write in accordance with the description details of Table 5-223 'XML Element Name
      requestType'"
      </p:requestType>
      <p:requestData>
      "Write in accordance with the description details of Table 5-223 'XML Element Name
      requestData'"
      </p:requestData>
      <p:requestDatatype>
      "Write in accordance with the description details of Table 5-223 'XML Element Name
      requestDatatype'"
      </p:requestDatatype>
      <p:responseType>
      "Write in accordance with the description details of Table 5-223 'XML Element Name
      responseType'"
      </p:responseType>
      <p:responseData>
      "Write in accordance with the description details of Table 5-223 'XML Element Name
      responseData'"
      </p:responseData>
      <p:responseDatatype>
      "Write in accordance with the description details of Table 5-223 'XML Element Name
      responseDatatype'"
      </p:responseDatatype>
      <p:errType>"Write in accordance with the description details of Table 5-223 'XML
      Element Name errType'"</p:errType>
      <p:errCodeRange>
      "Write in accordance with the description details of Table 5-223 'XML Element Name
      errCodeRange'"
      </p:errCodeRange>
      <p:relatedMessage>
      "Write in accordance with the description details of Table 5-223 'XML Element Name
      relatedMessage'"
      </p:relatedMessage>
      <p:comment>"Write in accordance with the description details of Table 5-223 'XML
      Element Name name'"</p:comment>
   </p:messageMember>
   <p:messageMember label="Write in accordance with the description details of Table 5-222
   'Attribute Name label'">
      :
      :
   </p:messageMember>
   :
   :
</p:message>
```

5.3.7.3 MESSAGE operation

When a communication service for which the data format is specified is executed by the utility software on a module, the service request data format to the module (RequestFrame), service response data format from the module at normal end (ResponseFrame), and error response data format from the module at error occurrence (ErrorFrame) are defined and the service is executed using the format.

This format information is developed into a model as MESSAGE. The following items are described in MESSAGE. For the details of these items, see Table 5-219 of Section 5.3.7.1.

Message

- TARGET
- REQUEST_TYPE
- REQUEST_DATA
- RESPONSE_TYPE
- RESPONSE_DATA
- ERR_TYPE

Figure 5-24 shows an operation image of a communication service based on the above items.

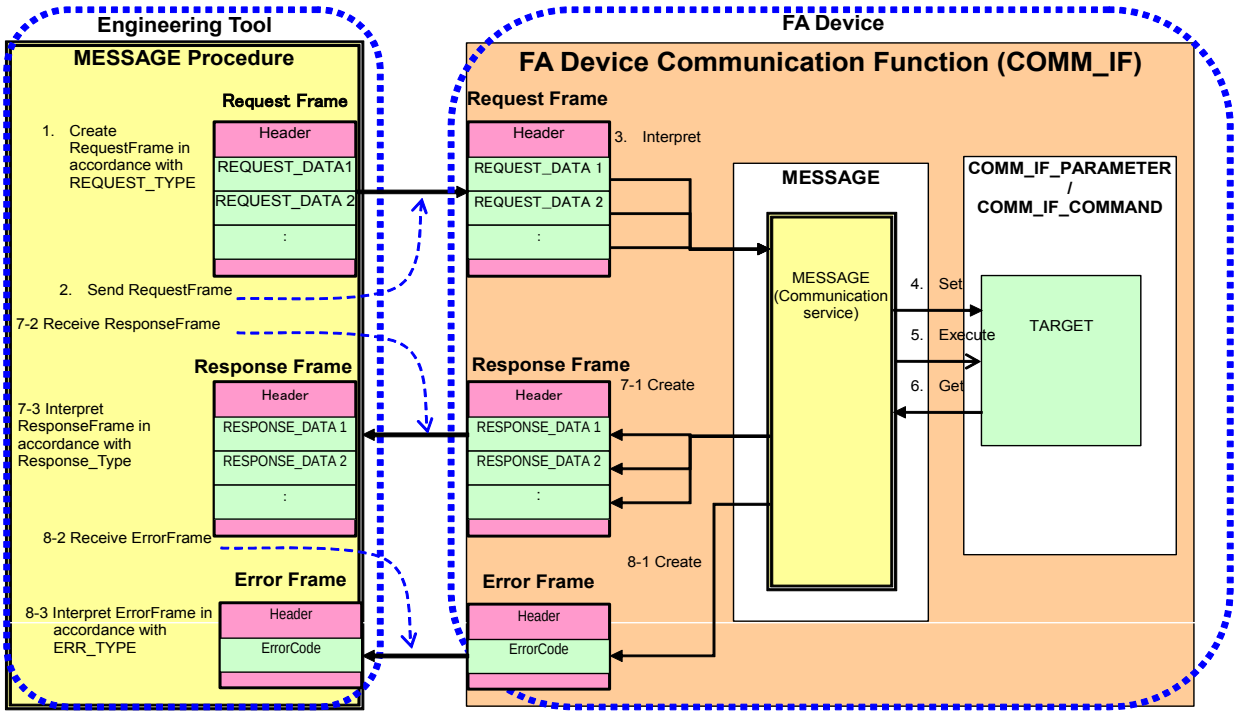


Figure 5-24 Operation of Communication Service Described in MESSAGE

The following shows the MESSAGE call-up and operation sequence. Figure 5-25 shows a sequence diagram of the sequence.

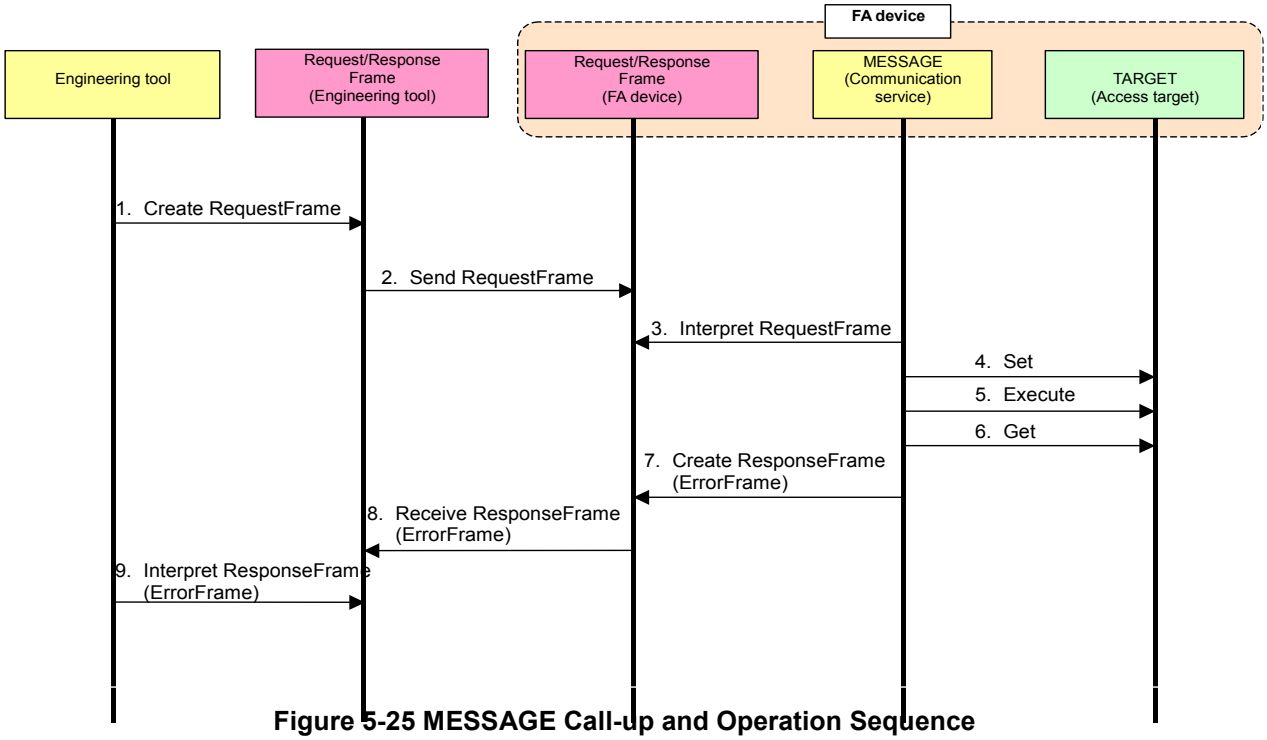


Figure 5-25 MESSAGE Call-up and Operation Sequence

(1) Create a RequestFrame of the data format determined by REQUEST_TYPE, and set REQUEST_DATA.

(2) Send the RequestFrame to the communication function of the FA device.

(3) The communication function of the FA device receives and passes the RequestFrame, reading the command code and setting value.

(4) Set the setting value in TARGET in accordance with the communication service and command code.

(5) Execute the processing of TARGET in accordance with the communication service and command code.

(6) Get the acquired value of TARGET in accordance with the communication service and command code.

[A. When a normal end occurs]

(7) Set the acquired value in the ResponseFrame in accordance with the data format corresponding to RESPONSE_TYPE, in accordance with the communication service and command code.

(8) Receive the ReseponseFrame from the communication function of the FA device.

(9) Interpret the ResponseFrame based on the data format determined by RESPONSE_TYPE, and read the RESPONSE_DATA.

[B. When an abnormal end occurs]

(7) Set the acquired value in the ErrorFrame in accordance with the data format corresponding to ERR_TYPE, in accordance with the communication service and command code.

(8) Receive the ErrorFrame from the communication function of the FA device.

(9) Interpret the ErrorFrame based on the data format determined by ERR_TYPE, and read the ErrorCode.

5.3.7.4 Description example of element

The following shows an item description example of an element of the MESSAGE part. In the example below, the element name is set to "SLMPGetParam".

Table 5-224 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "SLMPGetParam".
2.	LABEL2	Write "SLMPGetParam".
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "Parameter Read".
5.	TARGET	Write "Prm.*."
6.	MESSAGE_TYPE	Write "PARAMETER".
7.	REQUEST_TYPE	Write "RdReqSt_Binary".
8.	REQUEST_DATA	Write "<0x0401><0x0000><$(ASSIGN)><$(DATATYPE).WORDSIZE>" (with XML file: "<0x0401><0x0000><$(ASSIGN)><$(DATATYPE).WORDSIZE>")".
9.	REQUEST_DATATYPE	Write "<WORD><WORD><DWORD><WORD>"(with XML file: "<WORD><WORD><DWORD><WORD>").
10.	RESPONSE_TYPE	Write "rdResST_Binary".
11.	RESPONSE_DATA	Write "$(VALUE)".
12.	RESPONSE_DATATYPE	Write "$(DATATYPE)".
13.	ERR_TYPE	Write "rdErr_Binary".
14.	ERR_CODERANGE	Write a reference to the ENUM part "ErrorCode".
15.	RELATED_MESSAGE	Write a reference to the element of the MESSAGE part that indicates the pre-processing of the element of the MESSAGE part. Omittable item. Not written.
16.	COMMENT	Omittable item. Not written.

(a) Item description example of element

```
<p:messageMember label="SLMPGetParam">
   <p:label2><p:item>SLMPGetParam</p:item></p:label2>
   <p:name><p:item>Parameter Read</p:item></p:name>
   <p:target><p:item>Prm.*</p:item></p:target>
   <p:messageType><p:item>PARAMETER</p:item></p:messageType>
   <p:requestType><p:item>rdReqST_Binary</p:item></p:requestType>
   <p:requestData><p:item>&lt;0x0401&gt;&lt;0x0000&gt;&lt;$(ASSIGN)&gt;&lt;$(DATATYPE).WO
RDSIZE&gt;</p:item></p:requestData>
   <p:requestDatatype><p:item>&lt;WORD&gt;&lt;WORD&gt;&lt;DWORD&gt;&lt;WORD&gt;</p:ite
   m></p:requestDatatype>
   <p:responseType><p:item>rdResST_Binary</p:item></p:responseType>
   <p:responseData><p:item>$(VALUE)</p:item></p:responseData>
   <p:responseDatatype><p:item>$(DATATYPE)</p:item></p:responseDatatype>
   <p:errType><p:item>rdErr_Binary</p:item></p:errType>
   <p:errCodeRange><p:enumRefItem>ErrorCode</p:enumRefItem></p:errCodeRange>
</p:messageMember>
```

5.3.8 STRUCT part

5.3.8.1 Overview

The STRUCT part (structure) describes information related to the structure of the inputs and outputs of multiple elements. A structure is used when an area is divided. Each of the elements that make up the structure needs to be assigned a consecutive address. When an element within the COMM_IF section refers to a description of a STRUCT part, the STRUCT part is described within the same COMM_IF section.

The elements that make up the STRUCT part are defined in accordance with the structure used by the target module.

The configuration of each element of the STRUCT part, that is, the items that should be described within an element, is the same. Figure 5-26 shows the configuration of an element of the STRUCT part.

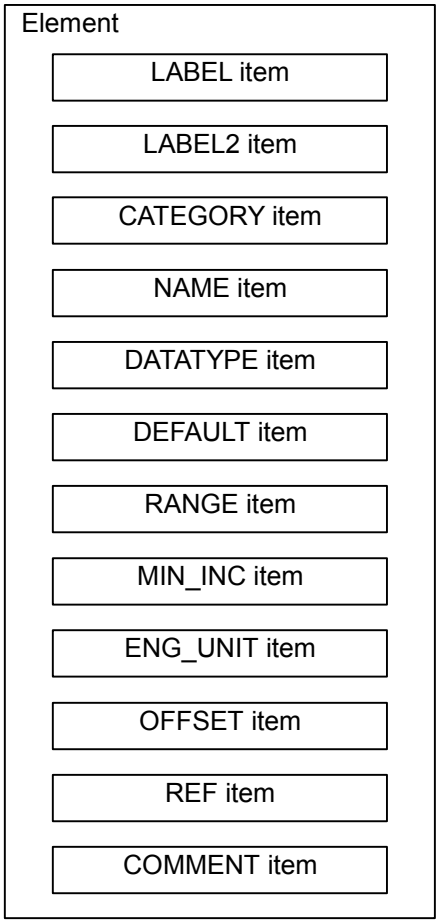


Figure 5-26 Structure Definition for Element of STRUCT Part

A single element is described as a single XML element. For the detailed description specifications of each element, see Section 5.3.8.3.

Table 5-225 describes the items included in an element.

Table 5-225 Item List for Elements Defined in STRUCT Part

No.	Item Name	Description Details	Description Specifications	Required /Optional
1.	LABEL	Describes the label for identifying the element	See Section 4.3.1.19	Required
2.	LABEL2	Describes the label for identifying the element. (Used when the utility software supports another language.)	See Section 4.3.1.20	Optional
3.	CATEGORY	Describes the category for grouping the element	See Section 4.3.1.4	Optional
4.	NAME	Describes the name of the element	See Section 4.3.1.24	Optional
5.	DATATYPE	Describes the data type of the element	See Section 4.3.1.10	Optional
6.	DEFAULT	Describes the default value set for the element	See Section 4.3.1.11	Optional
7.	RANGE	Describes the setting range of the element	See Section 4.3.1.26	Optional
8.	MIN_INC	Describes the minimum increment applied to the value of the element	See Section 4.3.1.23	Optional
9.	ENG_UNIT	Describes the engineering unit applied to the value of the element	See Section 4.3.1.13	Optional
10.	OFFSET	Describes the offset position of the element	See Section 4.3.1.25	Required
11.	REF	Describes the reference to be referred to by the element. Used to refer to the input/output of a BLOCK section from each element of the structure when the structure is defined in the COMM_IF section. * For describable references, see Section 4.3.1.30.	See Section 4.3.1.30	Optional
12.	COMMENT	Describes the meaning of the element as well as usage precautions.	See Section 4.3.1.6	Optional

5.3.8.2 XML format of STRUCT part

In XML format, the STRUCT part is described as an XML element having the XML element name "struct". Table 5-226 shows a list of the attributes of the struct element, and Table 5-227 shows a list of the XML child elements.

Table 5-226 Attribute List for struct Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the STRUCT part. (See Section 4.3.1.19.)	Required

Table 5-227 XML Child Element List for struct Element

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the STRUCT part. (See Section 4.3.1.6.)	Optional
2.	structMember	Describes the contents of the elements of the STRUCT part.	Required

In XML format, an element of the STRUCT part is described as an XML element having the XML element name "structMember". Table 5-228 shows a list of attributes of the structMember element, and Table 5-229 shows a list of XML child elements.

Table 5-228 Attribute List for structMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element. (See Section 4.3.1.19.)	Required

Table 5-229 XML Child Element List of structMember Element

No.	XML Element Name	Description Details
1.	label2	Describes the LABEL2 item
2.	category	Describes the CATEGORY item
3.	name	Describes the NAME item
4.	datatype	Describes the DATATYPE item
5.	default	Describes the DEFAULT item
6.	range	Describes the RANGE item
7.	minInc	Describes the MIN_INC item
8.	engUnit	Describes the ENG_UNIT item
9.	offset	Describes the OFFSET item
10.	ref	Describes the REF item
11.	comment	Describes the COMMENT item

Write the STRUCT part using a format such as shown below.

```
<p:struct label=" Write in accordance with the description details of Table 5-226 'Attribute
Name label'">
   <p:comment>"Write in accordance with the description details of Table 5-227 'XML Element
   Name comment'"</p:comment>
   <p:structMember label="Write in accordance with the description details of Table 5-228
   'Attribute Name label'">
      <p:label2>"Write in accordance with the description details of Table 5-229 'XML Element
      Name label2'">
      <p:category>"Write in accordance with the description details of Table 5-229 'XML
      Element Name category'"</p:category>
      <p:name>"Write in accordance with the description details of Table 5-229 'XML Element
      Name name'"</p:name>
      <p:datatype>"Write in accordance with the description details of Table 5-229 'XML
      Element Name datatype'"</p:datatype>
      <p:default>"Write in accordance with the description details of Table 5-229 'XML
      Element Name default'"</p:default>
      <p:range>"Write in accordance with the description details of Table 5-229 'XML Element
      Name range'"</p:range>
      <p:minInc>"Write in accordance with the description details of Table 5-229 'XML Element
      Name minInc'"</p:minInc>
      <p:engUnit>"Write in accordance with the description details of Table 5-229 'XML
      Element Name engUnit'"</p:engUnit>
      <p:offset>"Write in accordance with the description details of Table 5-229 'XML Element
      Name offset'"</p:offset>
      <p:ref>"Write in accordance with the description details of Table 5-229 'XML Element
      Name ref'"</p:ref>
      <p:comment>"Write in accordance with the description details of Table 5-229 'XML
      Element Name comment'"</p:comment>
   </p:structMember>
   <p:structMember label="Write in accordance with the description details of Table 5-228
   'Attribute Name labe'l">
      :
      :
   </p:structMember>
   :
   :
</p:struct>
```

5.3.8.3 Description example of element

The following shows item description examples of an element of the STRUCT part. In the examples, the element names are set to "ConvertEnable1", "ConvertEnable2", "ConvertEnable3", and "ConvertEnable4", respectively.

Table 5-230 Item Description Example of Element (ConvertEnable1)

No.	Item Name	Description Details
1.	LABEL	Write "ConvertEnable1"
2.	LABEL2	Write "ConvertEnable1"
3.	NAME	Write "Convert Enable CH1"
4.	DATATYPE	Write "BOOL"
5.	OFFSET	Write "0.0"
6.	REF	Write "BlockCH1.BlockInput.ConvertEnable"

Table 5-231 Item Description Example of Element (ConvertEnable2)

No.	Item Name	Description Details
1.	LABEL	Write "ConvertEnable2"
2.	LABEL2	Write "ConvertEnable2"
3.	NAME	Write "Convert Enable CH2"
4.	DATATYPE	Write "BOOL"
5.	OFFSET	Write "0.1"
6.	REF	Write "BlockCH2.BlockInput.ConvertEnable"

Table 5-232 Item Description Example of Element (ConvertEnable3)

No.	Item Name	Description Details
1.	LABEL	Write "ConvertEnable3"
2.	LABEL2	Write "ConvertEnable3"
3.	NAME	Write "Convert Enable CH3"
4.	DATATYPE	Write "BOOL"
5.	OFFSET	Write "0.2"
6.	REF	Write "BlockCH3.BlockInput.ConvertEnable"

Table 5-233 Item Description Example of Element (ConvertEnable4)

No.	Item Name	Description Details
1.	LABEL	Write "ConvertEnable4"
2.	LABEL2	Write "ConvertEnable4"
3.	NAME	Write "Convert Enable CH4"
4.	DATATYPE	Write "BOOL"
5.	OFFSET	Write "0.3"
6.	REF	Write "BlockCH4.BlockInput.ConvertEnable"

(a) Item description example of STConvertEnable element

```
<p:struct label="STConvertEnable">
   <p:structMember label="ConvertEnable1">
      <p:label2><p:item>ConvertEnable1</p:item></p:label2>
      <p:name><p:item>Convert Enable CH1</p:item></p:name>
      <p:datatype><p:item>BOOL</p:item></p:datatype>
      <p:offset><p:item>0.0</p:item></p:offset>
      <p:ref><p:item>BlockCH1.BlockInput.ConvertEnable</p:item></p:ref>
   </p:structMember>
   <p:structMember label="ConvertEnable2">
      <p:label2><p:item>ConvertEnable2</p:item></p:label2>
      <p:name><p:item>Convert Enable CH2</p:item></p:name>
      <p:datatype><p:item>BOOL</p:item></p:datatype>
      <p:offset><p:item>0.1</p:item></p:offset>
      <p:ref><p:item>BlockCH2.BlockInput.ConvertEnable</p:item></p:ref>
   </p:structMember>
   <p:structMember label="ConvertEnable3">
      <p:label2><p:item>ConvertEnable3</p:item></p:label2>
      <p:name><p:item>Convert Enable CH3</p:item></p:name>
      <p:datatype><p:item>BOOL</p:item></p:datatype>
      <p:offset><p:item>0.2</p:item></p:offset>
      <p:ref><p:item>BlockCH3.BlockInput.ConvertEnable</p:item></p:ref>
   </p:structMember>
   <p:structMember label="ConvertEnable4">
      <p:label2><p:item>ConvertEnable4</p:item></p:label2>
      <p:name><p:item>Convert Enable CH4</p:item></p:name>
      <p:datatype><p:item>BOOL</p:item></p:datatype>
      <p:offset><p:item>0.3</p:item></p:offset>
      <p:ref><p:item>BlockCH4.BlockInput.ConvertEnable</p:item></p:ref>
   </p:structMember>
</p:struct>
```

5.3.9 ENUM part

5.3.9.1 Overview

The ENUM part (option list) describes information related to options for values and return values to be set for an element. Refer to an ENUM part with elements in which you want to set settings based on a list box or elements for which you want to display the meaning of each value at the time a value is read using the utility software. If you want to refer to an ENUM part from an element within a COMM_IF section, write the ENUM part in the same COMM_IF section.

The elements that make up the ENUM part are defined in accordance with the options of the value to be used by the target module.

The configuration of each element of the ENUM part, that is, the items that should be described within an element, is the same. Figure 5-27 shows the configuration of an element of the ENUM part.

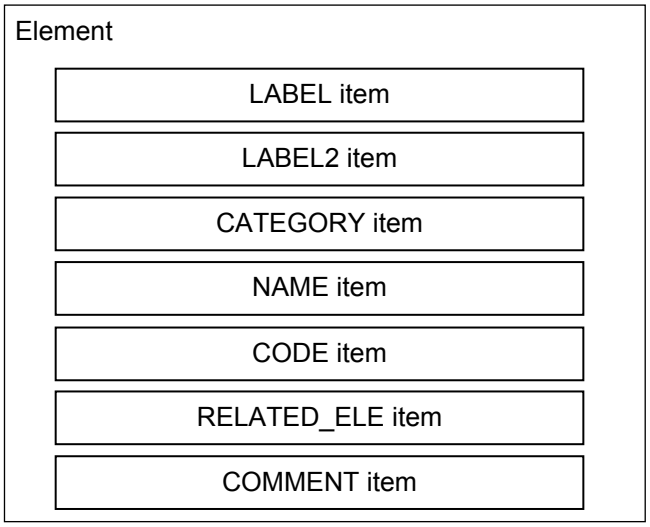


Figure 5-27 Structure Definition for Element of ENUM Part

A single element is described as a single XML element. For the detailed description specifications of each element, see Section 5.3.9.3.

Table 5-234 describes the items included in an element.

Table 5-234 Item List for Elements Defined in ENUM Part

No.	Item Name	Description Details	Description Specifications	Required /Optional
1.	LABEL	Describes the label for identifying the element	See Section 4.3.1.19	Required
2.	LABEL2	Describes the label for identifying the element. (Used when the utility software supports another language.)	See Section 4.3.1.20	Optional
3.	CATEGORY	Describes the category for grouping the element.	See Section 4.3.1.4	Optional
4.	NAME	Describes the name of the element. Used when the name and contents of the element are displayed with the utility software.	See Section 4.3.1.24	Required
5.	CODE	Describes the value for identifying the element. Cross-checks the value indicated by the element of the reference source with the value of the code, and selects the matching element.	See Section 4.3.1.5	Required
6.	RELATED_ELE	Describes the element information related to elements by referring to the COMMAND_ARGUMENT part. **[Note]** This is available only in CC-Link IE TSN.	See Section 4.3.1.31	Optional
7.	COMMENT	Describes the meaning of the element as well as usage precautions. **[Note]** In the case of CC-Link IE TSN, multiple COMMENT can be described. COMMENT1 and COMMENT2 are used.	See Section 4.3.1.6	Optional

5.3.9.2 XML format of ENUM part

In XML format, the ENUM part is described as an XML element having the XML element name "enum". Table 5-235 shows a list of the attributes of the enum element, and Table 5-236 shows a list of the XML child elements.

Table 5-235 Attribute List for enum Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the ENUM part. (See Section 4.3.1.19.)	Required

Table 5-236 XML Child Element List for enumElement

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the ENUM part. (See Section 4.3.1.6.)	Optional
2.	enumMember	Describes the contents of the elements of the ENUM part.	Required

In XML format, an element of the ENUM part is described as an XML element having the XML element name "enumMember". Table 5-237 shows a list of attributes of the enumMember element, and Table 5-238 shows a list of XML child elements. Table 5-239 shows a list of the XML child elements for CC-Link IE TSN.

Table 5-237 Attribute List for enumMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element. (See Section 4.3.1.19.)	Required

Table 5-238 XML Child Element List of enumMember Element

No.	XML Element Name	Description Details
1.	label2	Describes the LABEL2 item
2.	category	Describes the CATEGORY item
3.	name	Describes the NAME item
4.	code	Describes the CODE item
5.	comment	Describes the COMMENT item

Write the ENUM part using a format such as shown below.

```
<p:enum label=" Write in accordance with the description details of Table 5-235 'Attribute
Name label'">
   <p:comment>"Write in accordance with the description details of Table 5-236 'XML Element
   Name comment'"</p:comment>
   <p:enumMember label=" Write in accordance with the description details of Table 5-237
   'Attribute Name label'">
      <p:label2>"Write in accordance with the description details of Table 5-238 'XML Element
      Name label2'"</p:label2>
      <p:category>"Write in accordance with the description details of Table 5-238 'XML
      Element Name category"</p:category>
      <p:name>"Write in accordance with the description details of Table 5-238 'XML Element
      Name name'"</p:name>
      <p:code>"Write in accordance with the description details of Table 5-238 'XML Element
      Name code'"</p:code>
      <p:comment>"Write in accordance with the description details of Table 5-238 'XML
      Element Name comment'"</p:comment>
   </p:enumMember>
   <p:enumMember label="Write in accordance with the description details of Table 5-237
   'Attribute Name label'">
      :
      :
   </p:enumMember>
   :
   :
</p:enum>
```

Table 5-239 XML Child Element List of enumMember Element (For CC-Link IE TSN)

No.	XML Element Name	Description Details
1.	label2	Writes LABEL2 item.
2.	category	Writes CATEGORY item.
3.	name	Writes NAME item.
4.	code	Writes CODE item.
5.	relatedEle	Writes RELATED_ELE item. **[Note]** Available only in CC-Link IE TSN.
6.	comment index="1"	Writes COMMENT1 item.
7.	comment index="2"	Writes COMMENT2 item.

Write the ENUM part using a format such as shown below.

```
<p:enum label="Write in accordance with the description details of Table 5-235 'Attribute
Name label'">
   <p:comment>"Write in accordance with the description details of Table 5-236 'XML Element
   Name comment'"</p:comment>
   <p:enumMember label="Write in accordance with the description details of Table 5-237
   'Attribute Name label'">
      <p:label2>"Write in accordance with the description details of Table 5-239 'XML Element
      Name label2'"</p:label2>
      <p:category>"Write in accordance with the description details of Table 5-239 'XML
      Element Name category'"</p:category>
      <p:name>"Write in accordance with the description details of Table 5-239 'XML Element
      Name name'"</p:name>
      <p:code>"Write in accordance with the description details of Table 5-239 'XML Element
      Name code'"</p:code>
      <p:relatedEle>"Write in accordance with the description details of Table 5-239 'XML
      Element Name relatedEle'"</p:relatedEle>
      <p:comment index="1">
         "Write in accordance with the description details of Table 5-239 'XML Element Name
         comment index="1"'"
      </p:comment>
      <p:comment index="2">
         "Write in accordance with the description details of Table 5-239 'XML Element Name
         comment index="2"'"
      </p:comment>
   </p:enumMember>
   <p:enumMember label="Write in accordance with the description details of Table 5-237
   'Attribute Name label'">
      :
      :
   </p:enumMember>
   :
   :
</p:enum>
```

The above describes the format for CC-Link IE TSN. A description example is shown in Table 5-244 of Section 5.3.9.3.

5.3.9.3 Description example of element

The following shows item description examples of an element of the ENUM part. In the examples, the element names are set to "SelCH1", "SelCH2", "SelCH3", and "SelCH4", respectively. In the example of CC-Link IE TSN, the element name is set to "enum2".

Table 5-240 Item Description Example of Element (SelCH1)

No.	Item Name	Description Details
1.	LABEL	Write "SelCH1"
2.	LABEL2	Write "SelCH1"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "SelCH1"
5.	CODE	Write "1"
6.	COMMENT	Write "bit0"

Table 5-241 Item Description Example of Element (SelCH2)

No.	Item Name	Description Details
1.	LABEL	Write "SelCH2"
2.	LABEL2	Write "SelCH2"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "SelCH2"
5.	CODE	Write "2"
6.	COMMENT	Write "bit1"

Table 5-242 Item Description Example of Element (SelCH3)

No.	Item Name	Description Details
1.	LABEL	Write "SelCH3"
2.	LABEL2	Write "SelCH3"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "SelCH3"
5.	CODE	Write "3"
6.	COMMENT	Write "bit2"

Table 5-243 Item Description Example of Element (SelCH4)

No.	Item Name	Description Details
1.	LABEL	Write "SelCH4"
2.	LABEL2	Write "SelCH4"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "SelCH4"
5.	CODE	Write "4"
6.	COMMENT	Write "bit3"

Table 5-244 Item Description Example of Element (enum2)

No.	Item Name	Description Details
1.	LABEL	Write "enum2".
2.	LABEL2	Write "enum2".
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "Error Outside Input Value Range"
5.	CODE	Write "0x105".
6.	RELATED_ELE	Write "InputError".
7.	COMMENT1	Write "A value larger than the maximum value set for the parameter has been input.".
8.	COMMENT2	Write "Review the input or review the parameter setting.".

[Note] This description example can be used with CC-Link IE TSN only.

(a) Item description example of element

```
<p:enum label="enumChannelSelect ">
   <p:enumMember label="SelCH1">
      <p:label2><p:item>SelCH1</p:item></p:label2>
      <p:name><p:item>SelCH1</p:item></p:name>
      <p:code><p:item>1</p:item></p:code>
      <p:comment><p:item>bit0</p:item></p:comment>
   </p:enumMember>
   <p:enumMember label="SelCH2">
      <p:label2><p:item>SelCH2</p:item></p:label2>
      <p:name><p:item>SelCH2</p:item></p:name>
      <p:code><p:item>2</p:item></p:code>
      <p:comment><p:item>bit1</p:item></p:comment>
   </p:enumMember>
   <p:enumMember label="SelCH3">
      <p:label2><p:item>SelCH3</p:item></p:label2>
      <p:name><p:item>SelCH3</p:item></p:name>
      <p:code><p:item>3</p:item></p:code>
      <p:comment><p:item>bit2</p:item></p:comment>
   </p:enumMember>
   <p:enumMember label="SelCH4">
      <p:label2><p:item>SelCH4</p:item></p:label2>
      <p:name><p:item>SelCH4</p:item></p:name>
      <p:code><p:item>4</p:item></p:code>
      <p:comment><p:item>bit3</p:item></p:comment>
   </p:enumMember>
</p:enum>

<p:enum label="ErrorCodeList">
   <p:enumMember label="enum2">
      <p:label2><p:item>enum2</p:item></p:label2>
      <p:name><p:item>Error Outside Input Value Range</p:item></p:name>
      <p:code><p:item>0x105</p:item></p:code>
      <p:relatedEle><p:item>InputError</p:item></p:relatedEle>
      <p:comment index="1">
         <p:item>A value larger than the maximum value set for the parameter has been input.
         </p:item>
      </p:comment>
      <p:comment index="2">
         <p:item>Review the input or review the parameter setting. </p:item>
      </p:comment>
   </p:enumMember>
</p:enum>
```

5.3.10 COMMAND_ARGUMENT part

5.3.10.1 Overview

The COMMAND_ARGUMENT part (command argument list) describes information related to command arguments. When you want to refer to a description in a COMMAND_ARGUMENT part from an element within the COMM_IF section, write the COMMAND_ARGUMENT part in the same COMM_IF section.

The configuration of each element of the COMMAND_ARGUMENT part, that is, the items that should be described within an element, is the same. Figure 5-28 shows the configuration of an element of the COMMAND_ARGUMENT part.

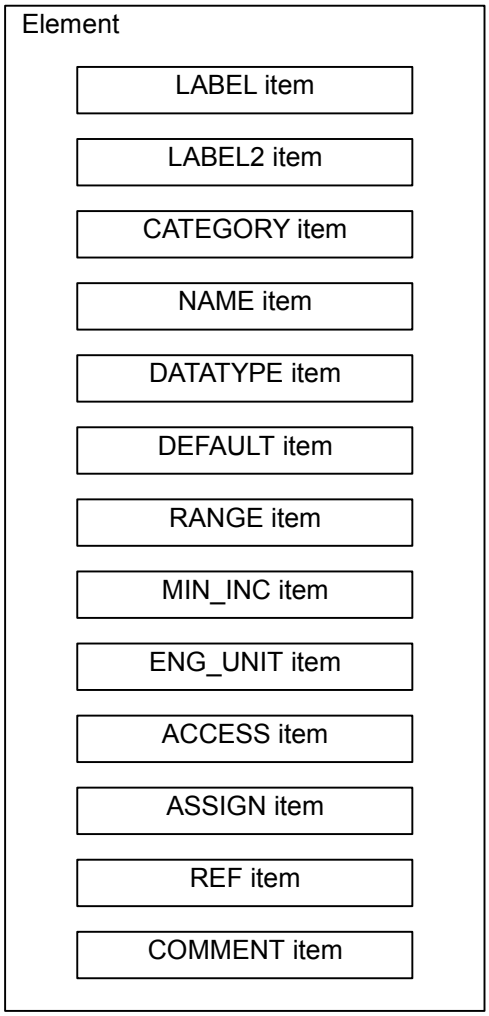


Figure 5-28 Structure Definition for Element of COMMAND_ARGUMENT Part

A single element is described as a single XML element. For the detailed description specifications of each element, see Section 5.3.10.3.

The items of the COMMAND_ARGUMENT part describe the argument information of a command.

Table 5-245 describes the items included in an element.

Table 5-245 Item List for Elements Defined in COMMAND_ARGUMENT Part

No.	Item Name	Description Details	Description Specifications	Required /Optional
1.	LABEL	Describes the label for identifying the element.	See Section 4.3.1.19	Required
2.	LABEL2	Describes the label for identifying the element. (Used when the utility software supports another language.)	See Section 4.3.1.20	Optional
3.	CATEGORY	Describes the category for grouping the element.	See Section 4.3.1.4	Optional
4.	NAME	Describes the name of the element. Used when the name and contents of the element are displayed with the utility software.	See Section 4.3.1.24	Required
5.	DATATYPE	Describes the data type of the element.	See Section 4.3.1.10	Required
6.	DEFAULT	Describes the setting value to be set for the element.	See Section 4.3.1.11	Optional
7.	RANGE	Describes the setting range of the element.	See Section 4.3.1.26	Optional
8.	MIN_INC	Describes the minimum increment applied to the value of the element of the command argument list along with ENG_UNIT.	See Section 4.3.1.23	Optional
9.	ENG_UNIT	Describes the engineering unit applied to the value of the element of the command argument list along with MIN_INC.	See Section 4.3.1.13	Optional
10.	ACCESS	Describes the access attribute of the element.	See Section 4.3.1.1	Required
11.	ASSIGN	Describes the address and code to be assigned to the element.	See Section 4.3.1.3	Optional
12.	REF	Describes the reference to be referred to by the element. Use prohibited under current specifications.	See Section 4.3.1.30	Optional
13.	COMMENT	Describes the meaning of the element as well as usage precautions.	See Section 4.3.1.6	Optional

5.3.10.2 XML format of COMMAND_ARGUMENT part

In XML format, the COMMAND_ARGUMENT part is described as an XML element having the XML element name "commandArgument". Table 5-246 shows a list of the attributes of the commandArgument element, and Table 5-247 shows a list of the XML child elements.

Table 5-246 Attribute List for commandArgument Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the COMMAND_ARGUMENT part. (See Section 4.3.1.19.)	Required

Table 5-247 XML Child Element List for commandArgumentElement

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the COMMAND_ARGUMENT part (See Section 4.3.1.6.)	Optional
2.	commIfInfoMember	Describes the contents of the elements of the COMMAND_ARGUMENT part	Required

In XML format, an element of the COMMAND_ARGUMENT part is described as an XML element having the XML element name "commandArgumentMember". Table 5-248 shows a list of attributes of the commandArgumentMember element, and Table 5-249 shows a list of XML child elements.

Table 5-248 Attribute List for commandArgumentMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element. (See Section 4.3.1.19.)	Required

Table 5-249 XML Child Element List of commandArgumentMember Element

No.	XML Element Name	Description Details
1.	label2	Describes the LABEL2 item
2.	category	Describes the CATEGORY item
3.	name	Describes the NAME item.
4.	datatype	Describes the DATATYPE item
5.	default	Describes the DEFAULT item
6.	range	Describes the RANGE item
7.	minInc	Describes the MIN_INC item.
8.	engUnit	Describes the ENG_UNIT item
9.	access	Describes the ACCESS item.
10.	assign	Describes the ASSIGN item.
11.	ref	Describes the REF item. Use prohibited under current specifications.
12.	comment	Describes the COMMENT item.

Write the COMMAND_ARGUMENT part using a format such as shown below.

```
<p:commandArgument label=" Write in accordance with the description details of Table 5-246
'Attribute Name label'">
   <p:comment>"Write in accordance with the description details of Table 5-247 'XML Element
   Name comment'"</p:comment>
   <p:commandArgumentMember label=" Write in accordance with the description details of
   Table 5-248 'Attribute Name label'">
      <p:label2>"Write in accordance with the description details of Table 5-249 'XML Element
      Name label2'"</p:label2>
      <p:category>"Write in accordance with the description details of Table 5-249 'XML
      Element Name category'"</p:category>
      <p:name>"Write in accordance with the description details of Table 5-249 'XML Element
      Name name'"</p:name>
      <p:datatype>"Write in accordance with the description details of Table 5-249 'XML
      Element Name datatype'"</p:datatype>
      <p:default>"Write in accordance with the description details of Table 5-249 'XML
      Element Name default'"</p:default>
      <p:range>"Write in accordance with the description details of Table 5-249 'XML Element
      Name range'"</p:range>
      <p:minInc>"Write in accordance with the description details of Table 5-249 'XML Element
      Name minInc'"</p:minInc>
      <p:engUnit>"Write in accordance with the description details of Table 5-249 'XML
      Element Name engUnit'"</p:engUnit>
      <p:access>"Write in accordance with the description details of Table 5-249 'XML Element
      Name access'"</p:access>
      <p:assign>"Write in accordance with the description details of Table 5-249 'XML Element
      Name assign'"</p:assign>
      <p:ref>"Write in accordance with the description details of Table 5-249 'XML Element
      Name ref'"</p:ref>
      <p:comment>"Write in accordance with the description details of Table 5-249 'XML
      Element Name comment'"</p:comment>
   </p:commandArgumentMember>
   <p:commandArgumentMember label=" Write in accordance with the description details of
   Table 5-248 'Attribute Name label'">
      :
      :
   </p:commandArgumentMember>
   :
   :
</p:commandArgument>
```

5.3.10.3 Description example of element

The following shows an item description example of an element of the COMMAND_ARGUMENT part. In the example below, the element name is set to "CH1_AvgOrSample".

Table 5-250 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "CH1_AvgOrSample"
2.	LABEL2	Write "CH1_AvgOrSample"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "CH1 Sampling and Averaging Process Specification"
5.	DATATYPE	Write "BOOL"
6.	DEFAULT	Write "0"
7.	RANGE	Omittable item. Not written.
8.	MIN_INC	Omittable item. Not written.
9.	ENG_UNIT	Omittable item. Not written.
10.	ACCESS	Write "W"
11.	ASSIGN	Omittable item. Not written.
12.	REF	Use prohibited
13.	COMMENT	Omittable item. Not written.

(a) Item description example of element

```
<p:commandArgumentMember label="CH1_AvgOrSample">
   <p:label2><p:item>CH1_AvgOrSample</p:item></p:label2>
   <p:name><p:item>CH1 Sampling and Averaging Process Specification</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:default><p:item>0</p:item></p:default>
   <p:access><p:item>W</p:item></p:access>
</p:commandArgumentMember>
```

5.4 BLOCK Section

The BLOCK section comprises multiple parts as shown below. One part is written as a single XML element. A comment may also be written (option).

BLOCK section	
BLOCK_INFO part	Describes function block identification information.
BLOCK_INPUT part	Describes function block input information.
BLOCK_OUTPUT part	Describes function block output information.
BLOCK_PARAMETER part	Describes function block parameter information.
BLOCK_COMMAND part	Describes the commands executable by the function block.
STRUCT part	Describes the structure of the inputs and outputs of multiple elements.
ENUM part	Describes the options for values to be set in the element and return values.
COMMAND_ARGUMENT part	Describes the argument information of BLOCK_COMMAND.

Figure 5-29 Structure of BLOCK Section

In XML format, the BLOCK section is written as an XML element of the XML element name "block". Table 5-251 shows a list of the attributes of the block element, and Table 5-252 shows a list of XML child elements.

Table 5-251 List of Attributes of block Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label that identifies the BLOCK section. (See Section 4.3.1.19.) (Example: "BlockSection")	Required

Table 5-252 List of XML Child Elements of block Element

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the BLOCK section (See Section 4.3.1.6.)	Optional
2.	blockInfo	Describes the contents of the BLOCK_INFO part	Required

5.4.1 BLOCK_INFO part

5.4.1.1 Overview

A BLOCK_INFO part describes information related to function block identification.

Figure 5-30 shows the elements that make up the BLOCK_INFO part.

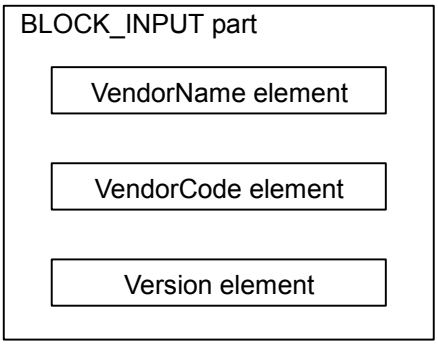


Figure 5-30 Structure of BLOCK_INFO Part

The configuration of each element of the BLOCK_INFO part, that is, the items that should be described within each element, is the same. Figure 5-31 shows the configuration of an element of the BLOCK_INFO part.

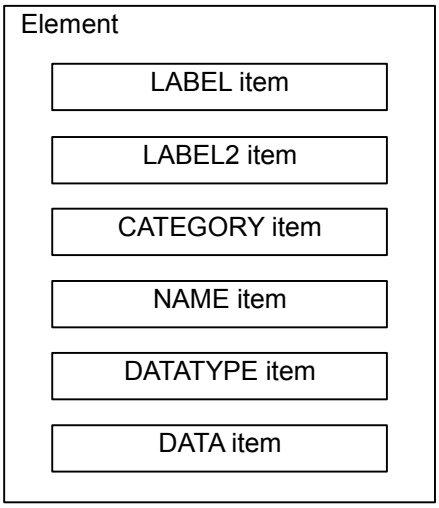


Figure 5-31 Structure Definition for Element of BLOCK_INFO Part

Table 5-253 lists the elements of the BLOCK_INFO part. A single element is described as a single XML element. For the detailed description specifications of each element, see Section 5.4.1.3.

Table 5-253 Elements of BLOCK Part

No.	Element	Description Details	Required/Optional
1.	VendorName	Describes the name of the vendor that manufactured the module.	Required
2.	VendorCode	Describes the vendor code of the vendor that manufactured the module. Describes the fifth to eighth digits of the partner member number of the CC-Link Partner Association.	Required
3.	Version	Describes the firmware version. Described using a string.	Required

Table 5-254 shows the items described in the elements of the BLOCK_INFO part.

Table 5-254 Item List of BLOCK_INFO Part

No.	Element Name	Description Details	Description Specifications	Required/Optional
1.	LABEL	Describes the label for identifying the element.	See Section 4.3.1.19	Required
2.	LABEL2	Describes the label for identifying the element. (Used when the utility software supports another language.)	See Section 4.3.1.20	Optional
3.	CATEGORY	Describes the category for grouping the element.	See Section 4.3.1.4	Optional
4.	NAME	Describes the name of the element. Used when the name and contents of the element are displayed with the utility software.	See Section 4.3.1.24	Optional
5.	DATATYPE	Describes the data type of the contents described in the DATA item.	See Section 4.3.1.10	Optional
6.	DATA	Describes the contents of the element.	See Section 4.3.1.9	Required

Based on the elements described in BLOCK_INFO, the information indicated in Table 5-253 is described as elements common to each device. Note, however, that LABEL2 and NAME are recommended specifications and may be omitted. For these items, we recommend writing "COMMON" for the CATEGORY.

In addition to the information shown in Table 5-253, information that is unique to the device may be added to BLOCK_INFO.

5.4.1.2 XML format of BLOCK_INFO part

In XML format, the BLOCK_INFO part is described as an XML element having the XML element name "blockInfo". Table 5-255 shows a list of the attributes of the blockInfo element, and Table 5-256 shows a list of the XML child elements.

Table 5-255 Attribute List for blockInfo Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the BLOCK_INFO part. (See Section 4.3.1.19.)	Required

Table 5-256 XML Child Element List for blockInfo Element

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the BLOCK_INFO part (See Section 4.3.1.6.)	Optional
2.	blockInfoMember	Describes the contents of the elements of the BLOCK_INFO part	Required

In XML format, the element of the BLOCK_INFO part is described as an XML element having the XML element name "blockInfoMember". Table 5-257 shows a list of attributes of the blockInfoMember element, and Table 5-258 shows a list of XML child elements.

Table 5-257 Attribute List for blockInfoMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element.(See Section 4.3.1.19.) The contents written for this label are regulated by CSP+ specifications, and cannot be arbitrarily written by the user. For details, see Section 5.4.1.3.	Required

Table 5-258 XML Child Element List of blockIfInfoMember Element

No.	XML Element Name	Description Details
1.	label2	Describes the LABEL2 item
2.	category	Describes the CATEGORY item
3.	name	Describes the NAME item
4.	datatype	Describes the DATATYPE item
5.	data	Describes the DATA item.

Write the BLOCK_INFO part using a format such as shown below.

```
<p:blockInfo label=" Write in accordance with the description details of Table 5-255
'Attribute Name label'">
   <p:comment>"Write in accordance with the description details of Table 5-256 'XML Element
   Name comment"</p:comment>
   <p:blockInfoMember label=" Write in accordance with the description details of Table
   5-257"Attribute Name label'">
      <p:label2>"Write in accordance with the description details of Table 5-258 'XML Element
      Name label2"</p:label2>
      <p:category>"Write in accordance with the description details of Table 5-258 'XML
      Element Name category"</p:category>
      <p:name>"Write in accordance with the description details of Table 5-258 'XML Element
      Name name"</p:name>
      <p:datatype>"Write in accordance with the description details of Table 5-258 'XML
      Element Name datatype"</p:datatype>
      <p:data>"Write in accordance with the description details of Table 5-258 'XML Element
      Name data"</p:data>
   </p:blockInfoMember>
   <p:blockInfoMember label=""Write in accordance with the description details of Table
   5-257'Attribute Name label'">
      :
      :
   </p:blockInfoMember>
   :
   :
</p:blockInfo>
```

5.4.1.3 Description details of each element

(1) Item description of VendorName element

Table 5-259 Item Description Specifications of VendorName Element

No.	Item Name	Description Details
1.	LABEL	Write "VendorName"
2.	LABEL2	Write "VendorName" in the case of English CSP+ files [Recommended specifications].
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "VendorName" in the case of English CSP+ files [Recommended specifications].
5.	DATATYPE	Write "STRING_U(64)"
6.	DATA	Write the name of the vendor that created the device

(a) Item description example of VendorName element

```
<p:blockInfoMember label="VendorName">
   <p:label2><p:item>VendorName</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Vendor Name</p:item></p:name>
   <p:datatype><p:item>STRING(64)</p:item></p:datatype>
   <p:data><p:item>CC-Link Partner Association</p:item></p:data>
</p:blockInfoMember>
```

(2) Item description of VendorCode element

Table 5-260 Item Description Specifications of VendorCode Element

No.	Item Name	Description Details
1.	LABEL	Write "VendorCode".
2.	LABEL2	Write "VendorCode" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON".
4.	NAME	Write "VendorCode" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Write "WORD".
6.	DATA	Write the code of the vendor that created the device. Write the fifth to eighth digits of the partner member number of the CC-Link Partner Association.

(a) Item description example of VendorCode element

```
<p:blockInfoMember label="VendorCode">
   <p:label2><p:item>VendorCode</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Vendor Code</p:item></p:name>
   <p:datatype><p:item>WORD</p:item></p:datatype>
   <p:data><p:item>0x0000</p:item></p:data>
</p:blockInfoMember>
```

(3) Item description of Version element

Table 5-261 Item Description Specifications of Version Element

No.	Item Name	Description Details
1.	LABEL	Write "Version"
2.	LABEL2	Write "Version" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Write "COMMON"
4.	NAME	Write "Version" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write a value from the following: bit string type, signed integer type, unsigned integer type, string type [expressed as "STRING(x) only". "STRING(32) is written in the example below.
6.	DATA	Write the version of the control function of the device. For the version method, see Section 5.2.1.1(1).

(a) Item description example of VendorName element

```
<p:blockInfoMember label="Version">
   <p:label2><p:item>Version</p:item></p:label2>
   <p:category><p:item>COMMON</p:item></p:category>
   <p:name><p:item>Version</p:item></p:name>
   <p:datatype><p:item>STRING(32)</p:item></p:datatype>
   <p:data><p:item>1.00A</p:item></p:data>
</p:blockInfoMember>
```

5.4.2 BLOCK_INPUT part

5.4.2.1 Overview

A BLOCK_INPUT part describes information related to function block input.

The elements that make up the BLOCK_INPUT part are defined in accordance with the function of the target module.

The configuration of each element of the BLOCK_INPUT part, that is, the items that should be described within each element, is the same. Figure 5-32 shows the configuration of an element of the BLOCK_INPUT part.

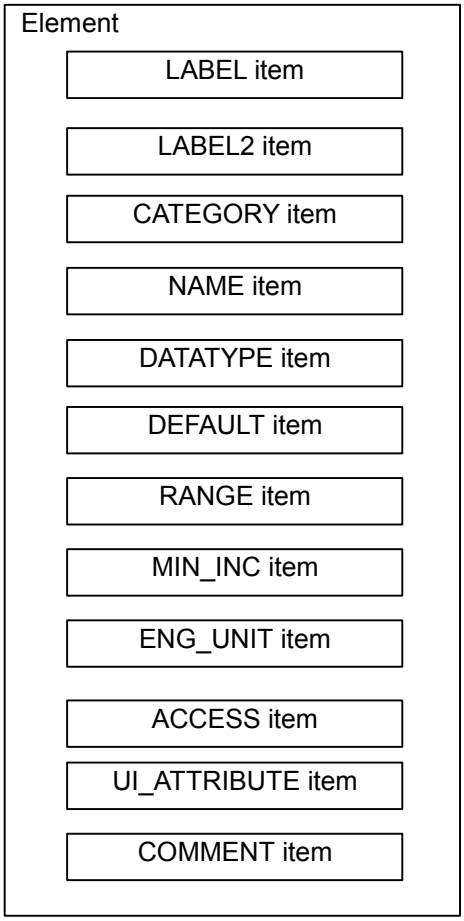


Figure 5-32 Structure Definition for Element of BLOCK_INPUT Part and BLOCK_OUTPUT Part

A single element is described as a single XML element. For element description examples, see Section 5.4.2.3.

Table 5-262 describes the items included in an element.

Table 5-262 List of Items Defined inBLOCK_INPUT Part and BLOCK_OUTPUT Part

No.	Element Name	Description Details	Description Specifications	Required /Optional
1.	LABEL	Describes the label for identifying the element. (Used when the utility software supports another language.)	See Section 4.3.1.19	Required
2.	LABEL2	Describes the label for identifying the element. (Used when the utility software supports another language.)	See Section 4.3.1.20	Optional
3.	CATEGORY	Describes the category for grouping the element.	See Section 4.3.1.4	Optional
4.	NAME	Describes the name of the element. Used when the name and contents are displayed with the utility software.	See Section 4.3.1.24	Required
5.	DATATYPE	Describes the data type of the contents.	See Section 4.3.1.10	Required
6.	DEFAULT	Describes the default value to be set for the element.	See Section 4.3.1.11	Optional
7.	RANGE	Describes the setting range of the element.	See Section 4.3.1.26	Optional
8.	MIN_INC	Describes the minimum increment applied to the value of the element along with "engUnit". Required when engUnit is described.	See Section 4.3.1.23	Optional
9.	ENG_UNIT	Describes the engineering unit applied to the value of the element along with "minInc. "	See Section 4.3.1.13	Optional
10.	ACCESS	Describes the access attribute of the element.	See Section 4.3.1.1	Optional
11.	UI_ATTRIBUTE	Describes the display method when the element is displayed with the utility software.	See Section 4.3.1.43	Optional
12.	COMMENT	Describes the meaning of the element as well as usage precautions.	See Section 4.3.1.6	Optional

5.4.2.2 XML format of BLOCK_INPUT part

In XML format, the BLOCK_INPUT part is described as an XML element having the XML element name "blockInput". Table 5-263 shows a list of the attributes of the blockInput element, and Table 5-264 shows a list of the XML child elements.

Table 5-263 Attribute List for blockInput Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the BLOCK_INPUT part. (See Section 4.3.1.19.)	Required

Table 5-264 XML Child Element List for blockInput Element

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the BLOCK_INPUT part. (See Section 4.3.1.6.)	Optional
2.	blockInputMember	Describes the contents of the elements of the BLOCK_INPUT part	Required

In XML format, the element of the BLOCK_INPUT part is described as an XML element having the XML element name "blockInputMember". Table 5-265 shows a list of attributes of the blockInputMember element, and Table 5-266 shows a list of XML child elements.

Table 5-265 Attribute List for blockInputMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element. (See Section 4.3.1.19.)	Required

Table 5-266 XML Child Element List of blockIfInputMember Element

No.	XML Element Name	Description Details
1.	label2	Describes the LABEL2 item.
2.	category	Describes the CATEGORY item.
3.	name	Describes the NAME item.
4.	datatype	Describes the DATATYPE item.
5.	default	Describes the DEFAULT item.
6.	range	Describes the RANGE item.
7.	minInc	Describes the MIN_INC item.
8.	engUnit	Describes the ENG_UNIT item
9.	access	Describes the ACCESS item.
10.	uiAttribute	Describes the UI_ATTRIBUTE item
11.	comment	Describes the COMMENT item.

Write the BLOCK_INPUT part using a format such as shown below.

```
<p:blockInput label=" Write in accordance with the description details of Table 5-263
'Attribute Name label'">
   <p:comment>"Write in accordance with the description details of Table 5-264 'XML Element
   Name comment'"</p:comment>
   <p:blockInputMember label=" Write in accordance with the description details of Table
   5-265"Attribute Name label'">
      <p:label2>"Write in accordance with the description details of Table 5-266 'XML Element
      Name label'2"</p:label2>
      <p:category>"Write in accordance with the description details of Table 5-266 'XML
      Element Name category'"</p:category>
      <p:name>"Write in accordance with the description details of Table 5-266 'XML Element
      Name name'"</p:name>
      <p:datatype>"Write in accordance with the description details of Table 5-266 'XML
      Element Name datatype'"</p:datatype>
      <p:default>"Write in accordance with the description details of Table 5-266 'XML
      Element Name default'"</p:default>
      <p:range>"Write in accordance with the description details of Table 5-266 'XML Element
      Name range'"</p:range>
      <p:minInc>"Write in accordance with the description details of Table 5-266 'XML Element
      Name minInc'"</p:minInc>
      <p:engUnit>"Write in accordance with the description details of Table 5-266 'XML
      Element Name engUnit'"</p:engUnit>
      <p:access>"Write in accordance with the description details of Table 5-266 'XML Element
      Name access'"</p:access>
      <p:uiAttribute>"Write in accordance with the description details of Table 5-266 'XML
      Element Name uiAttribute'"</p:uiAttribute>
      <p:comment>"Write in accordance with the description details of Table 5-266 'XML
      Element Name comment'"</p:comment>
   </p:blockInputMember>
   <p:blockInputMember label="Write in accordance with the description details of Table
   5-265'Attribute Name label'">
      :
      :
   </p:blockInputMember>
   :
   :
</p:blockInput>
```

5.4.2.3 Description example of element

The following shows an item description example of an element of the BLOCK_INPUT part. In the example below, the element name is set to "CH1_ADConversionComp".

Table 5-267 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "CH1_ADConversionComp"
2.	LABEL2	Write "CH1_ADConversionComp" in the case of English CSP+ files
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "CH1 A/D Conversion Complete Flag" in the case of English CSP+ files
5.	DATATYPE	Write "BOOL"
6.	DEFAULT	Omittable item. Not written.
7.	RANGE	Omittable item. Not written.
8.	MIN_INC	Omittable item. Not written.
9.	ENG_UNIT	Omittable item. Not written.
10.	ACCESS	Omittable item. Not written.
11.	UI_ATTRIBUTE	Omittable item. Not written.
12.	COMMENT	Omittable item. Not written.

(a) Item description example of element

```
<p:blockInputMember label="CH1_ADConversionComp">
   <p:label2><p:item>CH1_ADConversionComp</p:item></p:label2>
   <p:name><p:item>CH1 A/D Conversion Complete Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
</p:blockInputMember>
```

5.4.3 BLOCK_OUTPUT part

5.4.3.1 Overview

A BLOCK_OUTPUT part describes information related to function block output.

The elements that make up the BLOCK_OUTPUT part are defined in accordance with the function of the target module.

The configuration of each element of the BLOCK_OUTPUT part, that is, the items that should be described within each element, is the same. For the configuration of an element of the BLOCK_OUTPUT part, see Figure 5-32 (same as BLOCK_INPUT).

A single element is described as a single XML element. For the detailed description specifications of each element, see Section 5.4.3.3.

Table 5-262 (same as BLOCK_INPUT) describes the items included in an element.

5.4.3.2 XML format of BLOCK_OUTPUT part

In XML format, the BLOCK_OUTPUT part is described as an XML element having the XML element name "blockOutput". Table 5-268 shows a list of the attributes of the blockOutput element, and Table 5-269shows a list of the XML child elements.

Table 5-268 Attribute List for blockOutput Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the BLOCK_OUTPUT part. (See Section 4.3.1.19.)	Required

Table 5-269 XML Child Element List for blockOutput Element

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the BLOCK_OUTPUT part. (See Section 4.3.1.6.)	Optional
2.	blockOutputMember	Describes the contents of the elements of the BLOCK_OUTPUT part.	Required

In XML format, the element of the BLOCK_OUTPUT part is described as an XML element having the XML element name "blockOutputMember". Table 5-270 shows a list of attributes of the blockOutputMember element, and Table 5-271 shows a list of XML child elements.

Table 5-270 Attribute List for blockOutputMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element. (See Section 4.3.1.19.)	Required

Table 5-271 XML Child Element List of blockOutputMember Element

No.	XML Element Name	Description Details
1.	label2	Describes the LABEL2 item.
2.	category	Describes the CATEGORY item.
3.	name	Describes the NAME item.
4.	datatype	Describes the DATATYPE item.
5.	default	Describes the DEFAULT item.
6.	range	Describes the RANGE item.
7.	minInc	Describes the MIN_INC item.
8.	engUnit	Describes the ENG_UNIT item
9.	access	Describes the ACCESS item.
10.	uiAttribute	Describes the UI_ATTRIBUTE item
11.	comment	Describes the COMMENT item.

Write the BLOCK_OUTPUT part using a format such as shown below.

```
<p:blockOutput label=" Write in accordance with the description details of Table 5-268
'Attribute Name label'">
   <p:comment>"Write in accordance with the description details of Table 5-269 'XML Element
   Name comment"</p:comment>
   <p:blockOutputMember label=" Write in accordance with the description details of Table
   5-270"Attribute Name label'">
      <p:label2>"Write in accordance with the description details of Table 5-271 'XML Element
      Name label2"</p:label2>
      <p:category>"Write in accordance with the description details of Table 5-271 'XML
      Element Name category"</p:category>
      <p:name>"Write in accordance with the description details of Table 5-271 'XML Element
      Name name"</p:name>
      <p:datatype>"Write in accordance with the description details of Table 5-271 'XML
      Element Name datatype"</p:datatype>
      <p:default>"Write in accordance with the description details of Table 5-271 'XML
      Element Name default"</p:default>
      <p:range>"Write in accordance with the description details of Table 5-271 'XML Element
      Name range"</p:range>
      <p:minInc>"Write in accordance with the description details of Table 5-271 'XML Element
      Name minInc"</p:minInc>
      <p:engUnit>"Write in accordance with the description details of Table 5-271 'XML
      Element Name engUnit"</p:engUnit>
      <p:access>"Write in accordance with the description details of Table 5-271 'XML Element
      Name access"</p:access>
      <p:uiAttribute>"Write in accordance with the description details of Table 5-271 'XML
      Element Name uiAttribute"</p:uiAttribute>
      <p:comment>"Write in accordance with the description details of Table 5-271 'XML
      Element Name comment"</p:comment>
   </p:blockOutputMember>
   <p:blockOutputMember label="Write in accordance with the description details of Table
   5-270 'Attribute Name label'">
      :
      :
   </p:blockOutputMember>
   :
   :
</p:blockOutput>
```

5.4.3.3 Description example of element

The following shows an item description example of an element of the BLOCK_OUTPUT part. In the example below, the element name is set to "VoltageCurrentSelect".

Table 5-272 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "VoltageCurrentSelect"
2.	LABEL2	Write "VoltageCurrentSelect"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "Voltage Current Selection"
5.	DATATYPE	Write "BOOL"
6.	DEFAULT	Omittable item. Not written.
7.	RANGE	Omittable item. Not written.
8.	MIN_INC	Omittable item. Not written.
9.	ENG_UNIT	Omittable item. Not written.
10.	ACCESS	Omittable item. Not written.
11.	UI_ATTRIBUTE	Omittable item. Not written.
12.	COMMENT	Write "0: Voltage, 1: Current".

(a) Item description example of element

```
<p:blockOutputMember label="VoltageCurrentSelect">
   <p:label2><p:item>VoltageCurrentSelect</p:item></p:label2>
   <p:name><p:item>Voltage Current Selection</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:comment><p:item>0: Voltage, 1: Current</p:item></p:comment>
</p:blockOutputMember>
```

5.4.4 BLOCK_PARAMETER part

5.4.4.1 Overview

A BLOCK_PARAMETER part describes information related to the parameters of the control function of the target module.

The elements that make up the BLOCK_OUTPUT part are defined in accordance with the target module communication function.

The configuration of each element of the BLOCK_PARAMETER part, that is, the items that should be described within each element, is the same. For the configuration of an element of the BLOCK_PARAMETER part, see Figure 5-33.

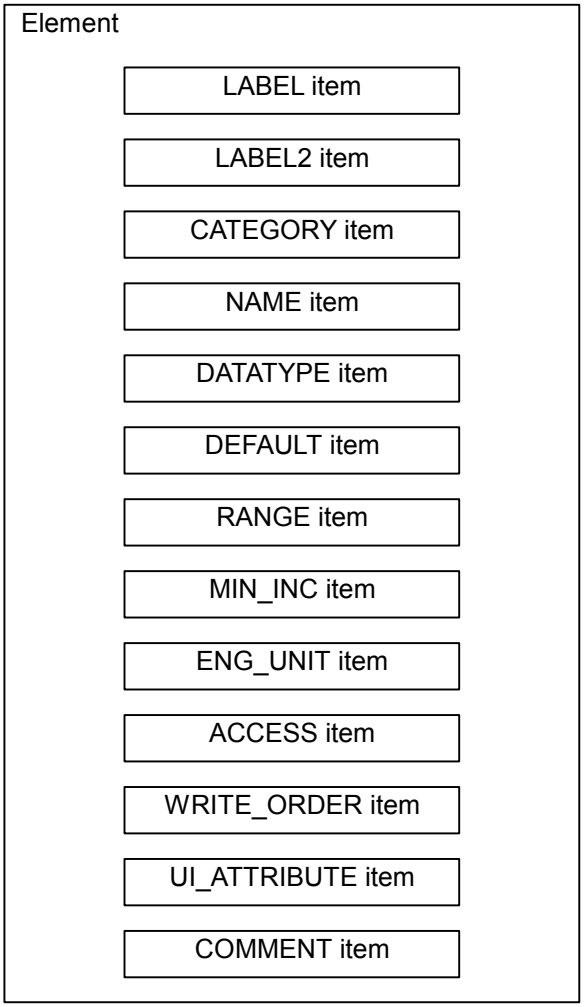


Figure 5-33 Structure Definition for Element of BLOCK_PARAMETER Part

A single element is described as a single XML element. For the detailed description specifications of each element, see Section 5.4.4.3.

An element of the BLOCK_PARAMETER part describes the parameters to be set in the module function area.

An element described in the BLOCK_PARAMETER part describes information about the parameters to be set in the function block.

Table 5-273 describes the items included in an element.

Table 5-273 Item List for Elements Defined in BLOCK_PARAMETER Part

No.	Item Name	Description Details	Description Specifications	Required /Optional
1.	LABEL	Describes the label for identifying the element.	See Section 4.3.1.19	Required
2.	LABEL2	Describes the label for identifying the element. (Used when the utility software supports another language.)	See Section 4.3.1.20	Optional
3.	CATEGORY	Describes the category for grouping the element.	See Section 4.3.1.4	Optional
4.	NAME	Describes the name of the element. Used when the name and contents are displayed with the utility software.	See Section 4.3.1.24	Required
5.	DATATYPE	Describes the data type of the element.	See Section 4.3.1.10	Required
6.	DEFAULT	Describes the default value set for the element.	See Section 4.3.1.11	Optional
7.	RANGE	Describes the setting range of the element.	See Section 4.3.1.26	Optional
8.	MIN_INC	Describes the minimum increment applied to the value of the element along with "engUnit".	See Section 4.3.1.23	Optional
9.	ENG_UNIT	Describes the engineering unit applied to the value of the element along with "minInc".	See Section 4.3.1.13	Optional
10.	ACCESS	Describes the access attribute of the element.	See Section 4.3.1.1	Required
11.	WRITE_ORDER	Describes the order in which elements are to be written to the module.	See Section 4.3.1.46	Optional
12.	UI_ATTRIBUTE	Describes the display method when the element is displayed with common utility software.	See Section 4.3.1.43	Optional
13.	COMMENT	Describes the meaning of the element as well as usage precautions.	See Section 4.3.1.6	Optional

5.4.4.2 XML format of BLOCK_PARAMETER part

In XML format, the BLOCK_PARAMETER part is described as an XML element having the XML element name "blockParameter". Table 5-274 shows a list of the attributes of the blockParameter element, and Table 5-275 shows a list of the XML child elements.

Table 5-274 Attribute List for blockParameter Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the BLOCK_PARAMETER part. (See Section 4.3.1.19.)	Required

Table 5-275 Child Element List for blockParameter Element

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the BLOCK_PARAMETER part. (See Section 4.3.1.6.)	Optional
2.	blockParameter Member	Describes the contents of the elements of the BLOCK_PARAMETER part.	Required

In XML format, the element of the BLOCK_PARAMETER part is described as an XML element having the XML element name "blockParameterMember". Table 5-276 shows a list of attributes of the blockParameterMember element, and Table 5-277 shows a list of child elements.

Table 5-276 Attribute List for blockParameterMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element. (See Section 4.3.1.19.)	Required

Table 5-277 Child Element List of blockParameterMember Element

No.	XML Element Name	Description Details
1.	label2	Describes the LABEL2 item.
2.	category	Describes the CATEGORY item.
3.	name	Describes the NAME item.
4.	datatype	Describes the DATATYPE item.
5.	default	Describes the DEFAULT item.
6.	range	Describes the RANGE item.
7.	minInc	Describes the MIN_INC item.
8.	engUnit	Describes the ENG_UNIT item
9.	access	Describes the ACCESS item.
10.	writeOrder	Describes the WRITE_ORDER item.
11.	uiAttribute	Describes the UI_ATTRIBUTE item
12.	comment	Describes the COMMENT item.

Write the BLOCK_PARAMETER part using a format such as shown below.

```
<p:blockParameter label="Write in accordance with the description details of Table 5-274
'Attribute Name label'">
   <p:comment>"Write in accordance with the description details of Table 5-275 'XML Element
   Name comment'"</p:comment>
   <p:blockParameterMember label=" Write in accordance with the description details of Table
   5-276 'Attribute Name label'">
      <p:label2>"Write in accordance with the description details of Table 5-277 'XML Element
      Name label2'"</p:label2>
      <p:category>"Write in accordance with the description details of Table 5-277 'XML
      Element Name category'"</p:category>
      <p:name>"Write in accordance with the description details of Table 5-277 'XML Element
      Name name'"</p:name>
      <p:datatype>"Write in accordance with the description details of Table 5-277 'XML
      Element Name datatype'"</p:datatype>
      <p:default>"Write in accordance with the description details of Table 5-277 'XML
      Element Name default'"</p:default>
      <p:range>"Write in accordance with the description details of Table 5-277 'XML Element
      Name range'"</p:range>
      <p:minInc>"Write in accordance with the description details of Table 5-277 'XML Element
      Name minInc'"</p:minInc>
      <p:engUnit>"Write in accordance with the description details of Table 5-277 'XML
      Element Name engUnit'"</p:engUnit>
      <p:access>"Write in accordance with the description details of Table 5-277 'XML Element
      Name access'"</p:access>
      <p:writeOrder>"Write in accordance with the description details of Table 5-277 'XML
      Element Name writeOrder'"</p:writeOrder>
      <p:uiAttribute>"Write in accordance with the description details of Table 5-277 'XML
      Element Name uiAttribute'"</p:uiAttribute>
      <p:comment>"Write in accordance with the description details of Table 5-277 'XML
      Element Name comment'"</p:comment>
   </p:blockParameterMember>
   <p:blockParameterMember label=" Write in accordance with the description details of Table
   5-276 'Attribute Name label'
      :
      :
   </p:blockParameterMember>
   :
   :
</p:blockParameter>
```

5.4.4.3 Description example of element

The following shows an item description example of an element of the BLOCK_PARAMETER part. In the example below, the element name is set to "Parameter1".

Table 5-278 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "Parameter1"
2.	LABEL2	Write "Parameter1"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "Parameter 1"
5.	DATATYPE	Write "UINT16"
6.	DEFAULT	Write "0"
7.	RANGE	Write " [0,65535]"
8.	MIN_INC	Write "0.01"
9.	ENG_UNIT	Write "Hz"
10.	ACCESS	Write "RW"
11.	WRITE_ORDER	Write "10"
12.	UI_ATTRIBUTE	Omittable item. Not written.
13.	COMMENT	Omittable item. Not written.

(a) Item description example of element

```
<p:blockParameterMember label="Parameter1">
   <p:label2><p:item>Parameter1</p:item></p:label2>
   <p:name><p:item>Parameter 1</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:default><p:item>0</p:item></p:default>
   <p:range><p:item>[0,65535]</p:item></p:range>
   <p:minInc><p:item>0.01</p:item></p:minInc>
   <p:engUnit><p:item>Hz</p:item></p:engUnit>
   <p:access><p:item>RW</p:item></p:access>
   <p:writeOrder><p:item>10</p:item></p:writeOrder>
</p:blockParameterMember>
```

5.4.5 BLOCK_COMMAND part

5.4.5.1 Overview

A BLOCK_COMMAND part describes information related to the commands to be executed by the control function of the target module.

The elements that make up the BLOCK_COMMAND part are defined in accordance with the function of the target module.

The configuration of each element of the BLOCK_COMMAND part, that is, the items that should be described within each element, is the same. For the configuration of an element of the BLOCK_COMMAND part, see Figure 5-34.

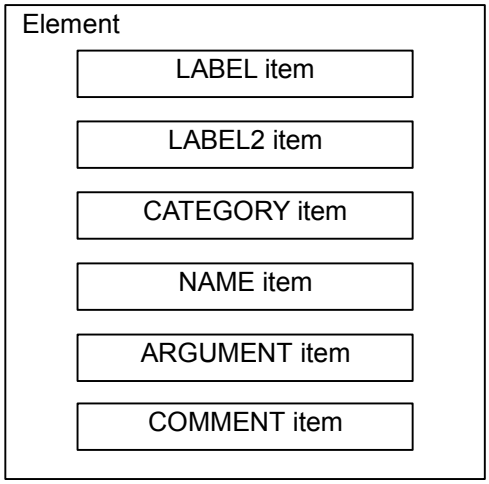


Figure 5-34 Structure Definition for Element of BLOCK_COMMAND Part

A single element is described as a single XML element. For the detailed description specifications of each element, see Section 5.4.5.3.

Table 5-279 shows the items described in an element of the BLOCK_COMMAND part.

Table 5-279 Item List for Elements Defined in BLOCK_COMMAND Part

No.	Item Name	Description Details	Description Specifications	Required /Optional
1.	LABEL	Describes the label for identifying the element.	See Section 4.3.1.19	Required
2.	LABEL2	Describes the label for identifying the element. (Used when the utility software supports another language.)	See Section 4.3.1.20	Optional
3.	CATEGORY	Describes the category for grouping the element.	See Section 4.3.1.4	Optional
4.	NAME	Describes the name of the item. Used when the name and contents are displayed with the utility software.	See Section 4.3.1.24	Required
5.	ARGUMENT	Describes the label of the COMMAND_ARGUMENT part for indicating the arguments used by the element.	See Section 4.3.1.3	Required
6.	COMMENT	Describes the meaning of the element as well as usage precautions.	See Section 4.3.1.6	Optional

5.4.5.2 XML format of BLOCK_COMMAND part

In XML format, the BLOCK_COMMAND part is described as an XML element having the XML element name "blockCommand". Table 5-280 shows a list of the attributes of the blockCommand element, and Table 5-281 shows a list of the XML child elements.

Table 5-280 Attribute List for blockCommand Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the BLOCK_COMMAND part. (See Section 4.3.1.19.)	Required

Table 5-281 XML Child Element List for blockCommand Element

No.	XML Element Name	Description Details	Required/Optional
1.	comment	Describes the comment related to the BLOCK_COMMAND part. (See Section 4.3.1.6.)	Optional
2.	blockCommand Member	Describes the contents of the elements of the BLOCK_COMMAND part.	Required

In XML format, the element of the BLOCK_COMMAND part is described as an XML element having the XML element name "blockCommandMember". Table 5-282 shows a list of attributes of the blockCommandMember element, and Table 5-283 shows a list of XML child elements.

Table 5-282 Attribute List for blockCommandMember Element

No.	Attribute Name	Description Details	Required/Optional
1.	label	Describes the label for identifying the element. (See Section 4.3.1.19.)	Required

Table 5-283 List of XML Child Elements of blockCommandMemberElement

No.	XML Element Name	Description Details	Required/Optional
1.	label2	Describes the label for identifying the element. (Used when the utility software supports another language.)	Optional
2.	category	Describes the category for grouping the element.	Optional
3.	name	Describes the name of the item. Used when the name and contents are displayed with the utility software.	Required
4.	argument	Describes the label of the COMMAND_ARGUMENT part for indicating the argument to be used by the element.	Required
5.	comment	Describes the meaning of the element as well as usage precautions.	Optional

Write the BLOCK_COMMAND part using a format such as shown below.

```
<p:blockCommand label="Write in accordance with the description details of Table 5-280
'Attribute Name label'">
   <p:comment>"Write in accordance with the description details of Table 5-281 'XML Element
   Name comment'"</p:comment>
   <p:blockCommandMember label="Write in accordance with the description details of Table
   5-282 'Attribute Name label'">
      <p:label2>"Write in accordance with the description details of Table 5-283 'XML Element
      Name label2'"</p:label2>
      <p:category>"Write in accordance with the description details of Table 5-283 'XML
      Element Name category'"</p:category>
      <p:name>"Write in accordance with the description details of Table 5-283 'XML Element
      Name name'"</p:name>
      <p:argument>"Write in accordance with the description details of Table 5-283 'XML
      Element Name argument'"</p:argument>
      <p:comment>"Write in accordance with the description details of Table 5-283 'XML
      Element Name comment'"</p:comment>
   </p:blockCommandMember>
   <p:blockCommandMember label="Write in accordance with the description details of Table
   5-282'Attribute Name label'">
      :
      :
   </p:blockCommandMember>
   :
   :
</p:blockCommand>
```

5.4.5.3 Description example of element

The following shows an item description example of an element of the BLOCK_COMMAND part. In the example below, the element name is set to "CmdGetFaultRecord".

Table 5-284 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "CmdGetFaultRecord"
2.	LABEL2	Write "CmdGetFaultRecord"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "Fault Record Acquisition"
5.	ARGUMENT	Write "ArgGetFaultRecord"
6.	COMMENT	Omittable item. Not written.

(a) Item description example of element

```
<p:blockCommandMember label="CmdGetFaultRecord">
   <p:label2><p:item>CmdGetFaultRecord</p:item></p:label2>
   <p:name><p:item>Fault Record Acquisition</p:item></p:name>
   <p:argument><p:item>ArgGetFaultRecord</p:item></p:argument>
</p:blockCommandMember >
```

5.4.6 STRUCT part

5.4.6.1 Overview

The STRUCT part (structure) describes information related to the structure of the inputs and outputs of multiple elements. A structure is used when an area is divided. Each of the elements that make up the structure needs to be assigned a consecutive address. When an element within the BLOCK section refers to a description of a STRUCT part, the STRUCT part is described within the same BLOCK section. A STRUCT part described in a BLOCK section defines the configuration of the elements described in the COMMAND_ARGUMEN part.

The elements that make up the STRUCT part are the same as those defined within a COMM_IF section. See Section 5.3.8. Note that when writing in a BLOCK section, the Name item and DATATYPE item are required, and the REF item is prohibited.

5.4.6.2 Description example of element

The following shows item description examples of an element of the STRUCT part. In the examples below, the element names are set to "STReturnCode1" and "STReturnCode2".

Table 5-285 Item Description Example of Element (STReturnCode1)

No.	Item Name	Description Details
1.	LABEL	Write "STReturnCode1".
2.	LABEL2	Write "STReturnCode1".
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "Reply Code 1".
5.	DATATYPE	Write "BYTE".
6.	DEFAULT	Omittable item. Not written.
7.	RANGE	Omittable item. Not written.
8.	MIN_INC	Omittable item. Not written.
9.	ENG_UNIT	Omittable item. Not written.
10.	OFFSET	Write "0.0".
11.	REF	Omittable item. Not written.
12.	COMMENT	Omittable item. Not written.

Table 5-286 Item Description Example of Element (STReturnCode2)

No.	Item Name	Description Details
1.	LABEL	Write "STReturnCode2".
2.	LABEL2	Write "STReturnCode2".
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "Reply Code 2".
5.	DATATYPE	Write "BYTE".
6.	DEFAULT	Omittable item. Not written.
7.	RANGE	Omittable item. Not written.
8.	MIN_INC	Omittable item. Not written.
9.	ENG_UNIT	Omittable item. Not written.
10.	OFFSET	Write "0.8".
11.	REF	Omittable item. Not written.
12.	COMMENT	Omittable item. Not written.

(a) Item description example of element

```
<p:struct label="STReturnCode">
   <p:structMember label="ReturnCode1">
   <p:label2><p:item>ReturnCode1</p:item></p:label2>
      <p:name><p:item>Reply Code 1</p:item></p:name>
      <p:datatype><p:item>BYTE</p:item></p:datatype>
      <p:offset><p:item>0.0</p:item></p:offset>
   </p:structMember>
   <p:structMember label="ReturnCode2">
      <p:label2><p:item>ReturnCode2</p:item></p:label2>
      <p:name><p:item>Reply Code 2</p:item></p:name>
      <p:datatype><p:item>BYTE</p:item></p:datatype>
      <p:offset><p:item>0.8</p:item></p:offset>
   </p:structMember>
</p:struct>
```

5.4.7 ENUM part

5.4.7.1 Overview

The ENUM part (option list) describes information related to the options of values to be set for an element. Refer to an ENUM part with elements in which you want to set settings based on a list box or elements for which you want to display the meaning of each value at the time a value is read using the utility software. If you want to refer to a description of an ENUM part from an element within a BLOCK section, write the ENUM part in the same BLOCK section.

The configuration of the ENUM part is the same as that defined for within the COMM_IF section. See Section 5.3.9.

5.4.7.2 Description example of element

The following shows item description examples of an element of the ENUM part. In the examples below, the element names are set to "PrmClear_CommPrmClear", "PrmClear_CommPrmNotClear", "AllPrmClear_CommPrmClear", and "AllPrmClear_CommPrmNotClear", respectively. In the example of CC-Link IE TSN, the element name is set to "enum2".

Table 5-287 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "PrmClear_CommPrmClear"
2.	LABEL2	Write "PrmClear_CommPrmClear"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "Parameter Clear (Communication Prameters Cleared)"
5.	CODE	Write "38550"
6.	COMMENT	Write "Command code: 9696h"

Table 5-288 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "PrmClear_CommPrmNotClear"
2.	LABEL2	Write "PrmClear_CommPrmNotClear"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "Parameter Clear (Communication Prameters Not Cleared)"
5.	CODE	Write "23130"
6.	COMMENT	Write "Command code: 5A5Ah"

Table 5-289 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "AllPrmClear_CommPrmClear"
2.	LABEL2	Write "AllPrmClear_CommPrmClear"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "All Parameter Clear (Communication Prameters Cleared)"
5.	CODE	Write "39270"
6.	COMMENT	Write "Command code: 9966h"

Table 5-290 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "AllPrmClear_CommPrmNotClear"
2.	LABEL2	Write "AllPrmClear_CommPrmNotClear"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "All Parameter Clear (Communication Prameters Not Cleared)"
5.	CODE	Write "21930".
6.	COMMENT	Write "Command code: 55AAh".

Table 5-291 Item Description Example of Element (enum2)

No.	Item Name	Description Details
1.	LABEL	Write "enum2".
2.	LABEL2	Write "enum2"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "Error Outside Input Value Range"
5.	CODE	Write "0x105".
6.	RELATED_ELE	Write "InputError".
7.	COMMENT1	Write "A value larger than the maximum value set for the parameter has been input.".
8.	COMMENT2	Write "Review the input or review the parameter setting.".

[Note] This description example can be used with CC-Link IE TSN only.

(a) Item description example of element

```
<p:enum label="enumPrmClearMode">
   <p:enumMember label="PrmClear_CommPrmClear">
      <p:label2><p:item>PrmClear_CommPrmClear</p:item></p:label2>
      <p:name><p:item>Parameter Clear (Communication Prameters Cleared)</p:item></p:name>
      <p:code><p:item>38550</p:item></p:code>
      <p:comment><p:item> Command code:9696h</p:item></p:comment>
   </p:enumMember>
   <p:enumMember label="PrmClear_CommPrmNotClear">
      <p:label2><p:item>PrmClear_CommPrmNotClear</p:item></p:label2>
      <p:name><p:item>Parameter Clear (Communication Prameters Not
      Cleared)</p:item></p:name>
      <p:code><p:item>23130</p:item></p:code>
      <p:comment><p:item> Command code:5A5Ah</p:item></p:comment>
   </p:enumMember>
   <p:enumMember label="AllPrmClear_CommPrmClear">
      <p:label2>
         <p:item>AllPrmClear_CommPrmClear</p:item>
      </p:label2>
      <p:name><p:item>All Parameter Clear (Communication Prameters
      Cleared)</p:item></p:name>
      <p:code><p:item>39270</p:item></p:code>
      <p:comment><p:item> Command code:9966h</p:item></p:comment>
   </p:enumMember>
   <p:enumMember label="AllPrmClear_CommPrmNotClear">
      <p:label2>
         <p:item>AllPrmClear_CommPrmNotClear</p:item>
      </p:label2>
      <p:name><p:item>All Parameter Clear (Communication Prameters Not
      Cleared)</p:item></p:name>
      <p:code><p:item>21930</p:item></p:code>
      <p:comment><p:item> Command code:55AAh</p:item></p:comment>
   </p:enumMember>
</p:enum>

<p:enum label="ErrorCodeList">
   <p:enumMember label="enum2">
      <p:label2>
         <p:item>enum2</p:item>
      </p:label2>
      <p:name><p:item>Error Outside Input Value Range</p:item></p:name>
      <p:code><p:item>0x105</p:item></p:code>
      <p:relatedEle><p:item>InputError</p:item></p:relatedEle>
      <p:comment index="1">
         <p:item>A value larger than the maximum value set for the parameter has been input.
         </p:item>
      </p:comment>
      <p:comment index="2">
         <p:item>Review the input or review the parameter setting. </p:item>
      </p:comment>
   </p:enumMember>
</p:enum>
```

5.4.8 COMMAND_ARGUMENT part

5.4.8.1 Overview

The COMMAND_ARGUMENT part (command argument list) describes information related to command arguments. When you want to refer to a description in a COMMAND_ARGUMENT part from an element within the BLOCK section, write the COMMAND_ARGUMENT part in the same BLOCK section.

The configuration of the COMMAND_ARGUMENT part is the same as that defined for within the COMM_IF section. See Section 5.3.10.

5.4.8.2 Description example of element

The following shows a description example for a case where the element is set to "PrmClearMode".

Table 5-292 Item Description Example of Element

No.	Item Name	Description Details
1.	LABEL	Write "PrmClearMode"
2.	LABEL2	Write "PrmClearMode"
3.	CATEGORY	Omittable item. Not written.
4.	NAME	Write "Prameter Clear Mode"
5.	DATATYPE	Write "UINT16"
6.	DEFAULT	Omittable item. Not written.
7.	RANGE	Write a reference to the ENUM part "enumPrmClearMode"
8.	MIN_INC	Omittable item. Not written.
9.	ENG_UNIT	Omittable item. Not written.
10.	ACCESS	Write "W"
11.	ASSIGN	Omittable item. Not written.
12.	REF	Omittable item. Not written.
13.	COMMENT	Omittable item. Not written.

(a) Item description example of element

```
<p:commandArgumentMember label="PrmClearMode">
   <p:label2><p:item>PrmClearMode</p:item></p:label2>
   <p:name><p:item>Prameter Clear Mode</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:range><p:enumRefItem>enumPrmClearMode</p:enumRefItem></p:range>
   <p:access><p:item>W</p:item></p:access>
</p:commandArgumentMember>
```

5.4.9 CONDITION Parts

5.4.9.1 Overview

In CONDITION Parts describes conditional expressions for when the value of items of elements change due to the current value of other elements. Conditional expressions describe reference to elements used for conditional determination and the condition specified range for those elements, as well as the results when it is true for each of the conditions specified range.

CONDITION Parts are, unlike the basic structure of parts shown in Section 4.2.3, composed of a single conditionVariable part and several conditionMemmber parts. The structure of CONDITION Parts is shown in Figure 5-35.

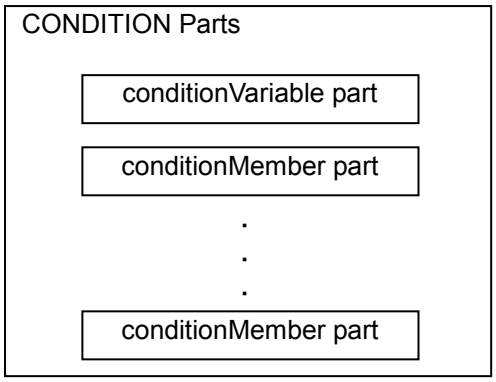


Figure 5-35 Structure of CONDITION Parts

The structure of conditionVariable parts of CONDITION Parts is shown in Figure 5-36, and the structure of conditionMemmber parts is shown in Figure 5-37.

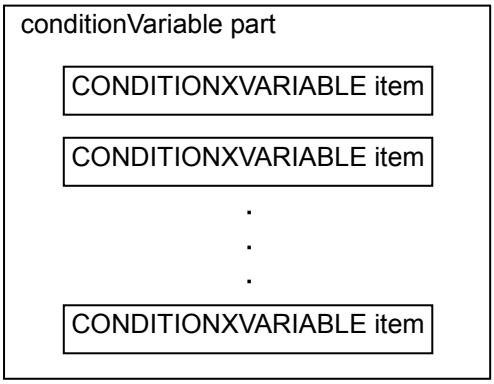


Figure 5-36 Structure of conditionVariable parts in CONDITION Parts

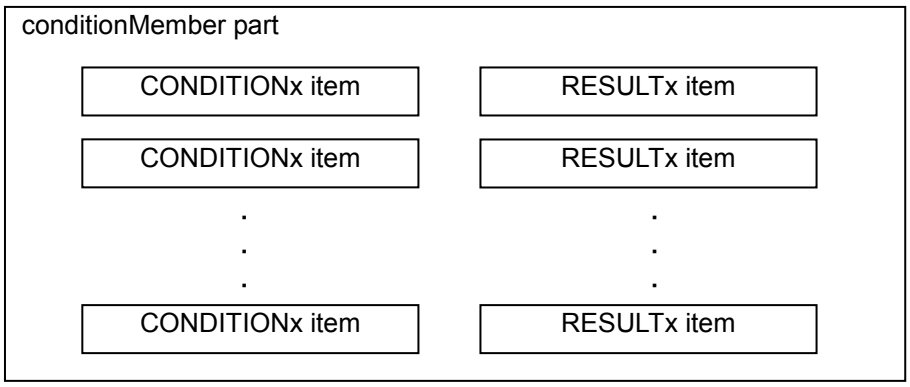


Figure 5-37 Structural definition of conditionMemmber parts in CONDITION Parts

Write 1 conditionVariable part/conditionMemmber part as 1 XML element. For the detailed description specifications of each element, see Section 5.4.9.3.

Table 5-293 List of conditionVariable parts of CONDITION Parts

No.	Element Name	Description Details	Required/Optional
1.	CONDITIONXVARIABLE	Writes reference to elements used for conditional judgment. When using multiple elements for conditional judgment, write multiple CONDITIONXVARIABLE items in conditionVariable parts.	Required

Table 5-294 List of conditionMember parts of CONDITION Parts

No.	Element Name	Description Details	Required/Optional
1.	CONDITIONx	Writes the range of conditions specified. If describing multiple ranges of conditions specified, it writes multiple CONDITIONx items in conditionMemmber parts.	Required
2	RESULTx	Writes the value applied when conditions are satisfied. If describing multiple results, it writes multiple RESULTx items in conditionMemmber parts.	Required

5.4.9.2 XML format of CONDITION Parts

In XML formats, CONDITION Parts are described as XML elements of XML element name "condition".

A list of condition element attributes is shown in Table 5-295, and a list of XML elements is shown in Table 5-296.

Table 5-295 List of condition element attributes

No.	Element Name	Description Details	Required/Optional
1.	label	Writes labels that identify CONDITION parts. (See 4.3.1.19) (Example: "ParamCondition1")	Required
2	conditionNumber	Writes the number of conditions used in conditional judgment with conditional expressions written in these CONDITION Parts. It writes as many CONDITIONXVARIABLE items/CONDITIONx items as are shown here in the conditionVariable parts/conditionMemmber parts in this CONDITION Part.	Required
3	resultNumber	Writes the number of results shown by conditional judgement with conditional expressions written in these CONDITION Parts. It shows as many RESULTx items as are shown here in the conditionMember parts in this CONDITION Part.	Required

Table 5-296 List of XML subelements of the condition element

No.	XMLElement Name	Description Details	Required/Optional
1.	comment	Writes comments on CONDITION Parts. (See 4.3.1.6)	Optional
2	conditionVariable	Writes references to elements used in conditional judgment of CONDITION Parts.	Required
3	conditionMember	Writes results that apply as specified range of conditions used for conditional judgement of CONDITION Parts.	Required

conditionVariable parts and conditionMemmber parts of CONDITION Parts are written as the XML elements "conditionVariable" or "conditionMember" in XML format. A list of XML subelements of conditionVariable parts are shown in Table 5-298. conditionVariable parts and conditionMemmber parts do not have such attributes as "label".

Table 5-297 List of XML subelements of conditionVariable parts

No.	XML Element Name	Description Details
1.	conditionXVariable	Writes CONDITIONXVARIABLE.

Table 5-298 List of XML subelements of conditionMember parts

No.	XML Element Name	Description Details
1.	conditionX	Writes CONDITIONx item.
2	resultX	Writes RESULTx item.

A list of the attributes of the conditionXVariable element, conditionX element, and resultX element are shown in Table 5-299 to Table 5-301.

Table 5-299 List of attributes of the conditionXVariable element

No.	Attribute Name	Description Details	Required/Optional
1.	index	Following the order of arrangement of multiple conditionXVariable elements written, writes integers in order from 1 to a maximum of 32.	Required

Table 5-300 List of attributes of the conditionXVariable element

No.	Attribute Name	Description Details	Required/Optional
1.	index	Writes the same value as the index value of the corresponding conditionXVariable element.	Required

Table 5-301 List of attributes of the conditionXVariable element

No.	Attribute Name	Description Details	Required/Optional
1.	index	Following the order of arrangement of multiple conditionXVariable elements written, writes integers in order from 1 to a maximum of 32.	Required

CONDITION parts are described in the following format.

```
<p:condition label="Write according to the details of Table 5-295 "Attribute label""
          conditionNumber="Write according to the details of Table 5-295 "Attribute
          conditionNumber""
          resultNumber="Write according to the details of Table 5-295 "Attribute
          resultNumber""">
  <p:comment>"Write according to the details of Table 5-296 "XML element comment""</p:comment>
  <p:conditionVariable>
    <p:conditionXVariable index="1">
      "Write according to the details of Table 5-297 "XML element conditionXVariable""
    </p: conditionXVariable>
    <p:conditionXVariable index="2">
      "Write according to the details of Table 5-297 "XML element conditionXVariable""
    </p: conditionXVariable>
      :
      :
  p:conditionVariable>
  <p:conditionMember>
    <p:conditionX index="1">
      "Write according to the details of Table 5-298 "XML element conditionX""
    </p: conditionX>
    <p:conditionX index="2">
      "Write according to the details of Table 5-298 "XML element conditionX""
    </p: conditionX>
      :
      :
    <p:resultX index="1">
      "Write according to the details of Table 5-298 "XML element resultX""
    </p:resultX>
    <p:resultX index="2">
      "Write according to the details of Table 5-298 "XML element resultX""
    </p:resultX>
      :
      :
  </p:conditionMember>
  <p:conditionMember>
    :
    :
  </p:conditionMember>
    :
    :
</p:condition>
```

5.4.9.3 Description details of elements

(1) Item descriptions of conditionVariable part

Table 5-302 Item description specifications of conditionVariable parts

No.	Element Name	Description Details
1.	CONDITIONXVARIABLE	Writes references to elements used for conditional judgment. When using multiple elements for conditional judgment, write multiple CONDITIONXVARIABLE items in conditionVariable parts. Writes references as follows. ・ Writes reference to the value used in conditional judgment. Cannot write reference to other items. ・ Cannot write references to structural type, array type, or constant type elements. ・ Cannot omit details of description. ・ Can only write reference to elements of BLOCK_PARAMETER parts. Can also write references elements of BLOCK_PARAMETER of parts that differ from the reference source. Cannot write references to elements of sections that differ from the reference source. ・ Cannot reference elements that apply results of this CONDITION part (value of RESULTx item) as elements used for conditional judgment of the reference source CONDITION part. In other words, it cannot write elements of BLOCK_PARAMETER parts that reference CONDITION parts as the reference source in CONDITIONXVARIABLE item of the same CONDITION parts.

Write a maximum of 32 CONDITIONXVARIABLE items in 1 conditionVariable part. In XML format, it has index as an attribute of conditionXVariable of XML elements, and grants integer values from 1 to 32 in the order those items are written.

(a) Example item description of conditionVariable part

```
<p:conditionVariable>
   <p:conditionXVariable index="1">
      <p:item>BlockParameter.AvgOrSample.VALUE</p:item>
   </p:conditionXVariable>
   <p:conditionXVariable index="2">
      <p:item>BlockParameter.TimeOrCount.VALUE</p:item>
   </p:conditionXVariable>
</p:conditionVariable>
```

(2) Item descriptions of conditionMemmber parts

Table 5-303 Item description specifications of conditionMemmber parts

No.	Element Name	Description Details
1.	CONDITIONx	Describes the range of conditions for conducting conditional judgment. ・ The following conventions are available as ranges of conditions. 1. Describing range of conditions with numerical values Specifies judgment conditions by numerical value. When enumerating multiple values, uses "," (commas), and when indicating the range of values, uses "[","]","(", and ")". (See [4] of Table 4-23) Ex: 2 ≤ x ≤ 10 ⇒ [2,10] 2 < x < 10 ⇒ (2,10) x < 10 ⇒ (,10) x = 10 or 20 ≤ x < 30 or x = 40 ⇒ 10,[20,30),40 2. Describing range of conditions using values of other elements Can perform conditional judgment with values of other elements of BLOCK_PARAMETER using reference source index by "$()" Ex: When judging whether a label name is below the value of the parameter of B, it can write this as [,$(B. VALUE)] Describing string type conditions 3. Writes the string being judged as a judgment condition. When judging string type, it is possible to specify conditions by perfect match with the string. Range cannot be specified. ・ Values usable in specifying the range of conditions shall be those values allowed by DATATYPE of elements used in conditional judgment (elements written in CONDITIONXVARIABLE item). (Ex: If the DATATYPE is word, hexadecimal numbers with "0x" can be used.) ・ If this item is omitted, corresponding values are not judged. ・ When using the fact that the value of the element referenced with the conditionVariable part has not been set (i.e., the item is not described), write the keyword "EMPTY". ・ In the final conditionMemmber part written, writes the keyword "DEFAULT" in all CONDITIONx items. CondtionMemmber parts written this way represent cases in which all conditional judgments were false. DEFAULT must be written. Therefore, in the final conditionMemmber part written in CONDITION part, be certain to perform description by this DEFAULT.
2	RESULTx	Write the values (results) applicable when conditions are satisfied. ・ When writing multiple results, write multiple RESULTx items in the conditionMemmber part. ・ Values written here must be writable in the reference source item referencing this item (the item these results are applied to).

A maximum of 32 CONDITIONx items and RESULTx items can be written in 1 conditionMemmber part. In XML format, they have index as attributes of the XML elements conditionX and ResultX, and grant integer values from 1 to 32 in the order those items are written. When indicating individual CONDITIONx items or RESULTx items, for convenience, replace the "x" part of those items with index values, and denote them as CONDITION1, CONDITION2, ~CONDITION32, etc. Also when writing a reference to RESULTx items in items of elements of BLOCK_PARAMETER, denote item names as RESULT1, RESULT2, ~RESULT32.

The condition range specifications written in CONDITIONx items indicate the range of values of elements referenced from CONDITIONXVARIABLE which holds an index of the same values as conditionVariable part.

Those CONDITIONx items correspond to the CONDITIONXVARIABLE items of the conditionVariable part which holds the same index values, and if the values of elements written in CONDITIONXVARIABLE item are within the range written in this item, judge it as satisfying the condition.

When multiple CONDITIONXVARIABLE items are written in the conditionVariable part, judge multiple conditions of those CONDITIONx items written in 1 conditionMemmber part as AND conditions. In other words, when all conditions of CONDITIONx items written in the conditionMemmber part are satisfied, the value of the RESULTx item written in that conditionMemmber part is applied as the result.

Conditional judgement is performed in the order written in conditionMemmber parts. Therefore, when it matches conditions written in multiple conditionMemmber parts, the results of the conditionMemmber part written first are applied. It is possible to write the same values for all RESULTx items in multiple conditionMemmber parts, and by changing the judgment conditions of CONDITIONx item, write judgment by OR condition of multiple judgment conditions.

(a) Example of item description of a conditionMemmber part

```
<p:conditionMember>
   <p:conditionX index="1">
      <p:item>1</p:item>
   </p:conditionX>
   <p:conditionX index="2">
      <p:item>0</p:item>
   </p:conditionX>
   <p:resultX index="1">
      <p:item>NA</p:item>
   </p:resultX>
   <p:resultX index="2">
      <p:item>RW</p:item>
   </p:resultX>
</p:conditionMember>
```

5.4.9.4 Writing conditional expressions

Conventions for conditional expressions of CONDITION parts are introduced here.

An example conditional expression is shown in Figure 5-38, and the relationship between conditional expression (CONDITION) and parameters are shown here (Figure 5-39).

In the example in Figure 5-38, the number of elements used in conditional judgement (reference source from CONDITIONXVARIABLE item) is 2, and the number of results from the conditional judgment (RESULTx) is also 2.

Row 1 has the details written of each item of conditionVariable part, and rows 2 onwards have the details written of each item of conditionMemmber part.

When the values of the element written in the CONDITIONXVARIABLE of conditionVariable in the first row all match the condition range written in CONDITIONx items of conditionMemmber parts in rows 2 onward, the values written in RESULTx of those conditionMemmber are applied. This result is applied by referencing from items of parameters written in BLOCK_PARAMETER.

C_AvgTimeCount

	CONDITION1	CONDITION2	RESULT1	RESULT2
1	BlockParameter.AvgOrSample.VALUE	BlockParameter.TimeOrCount.VALUE		
2	0		NA	NA
3	1	0	NA	RW
4	1	1	RW	NA
5	DEFAULT	DEFAULT	NA	NA

Figure 5-38 Example of conditional expression (CONDITION)

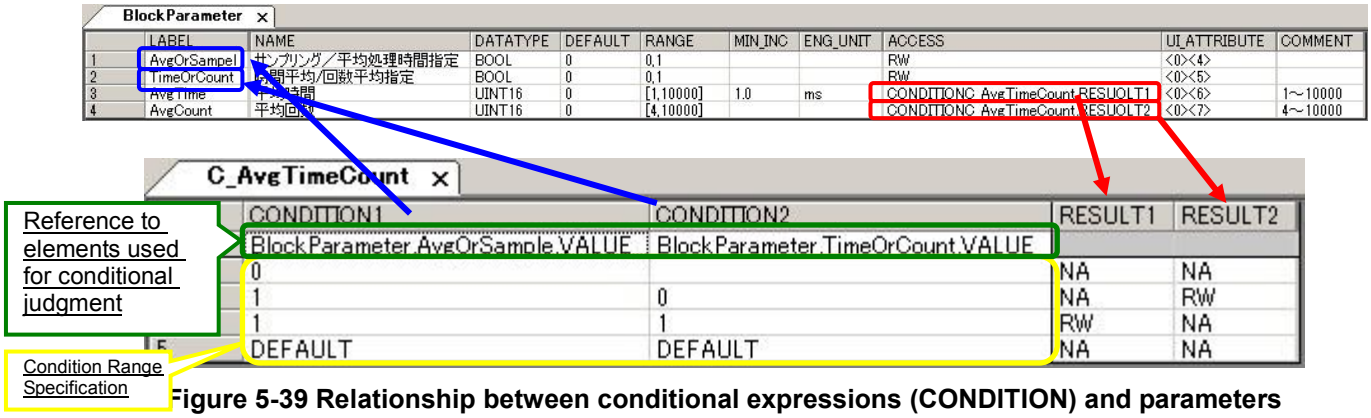


Figure 5-39 Relationship between conditional expressions (CONDITION) and parameters

Related items are written about below.

(1) References to elements used in conditional judgment (CONDITIONXVARIABLE items of conditionVariable parts)

Specify the value used for conditional judgment by writing a reference to the values of the element used for conditional judgment in CONDITIONXVARIABLE items of conditionVariable parts. See Section 5.4.9.3 (1) for details about this convention.

(2) Condition range specification (CONDITIONx items of conditionMemmber parts)

Write judgment conditions of the value specified in (1) above using numerical values, strings, or other elements. If the value is within the range written in this item it is judge as satisfying the condition. See Section 5.4.9.3 (2) for details about this convention.

(3) Result (RESULTx item of conditionMemmber parts)

The value of the results are applied as the values of the item that references this result. Therefore, the values written here must be values writable with items of the reference source. In the example in Figure 5-39, because both results are referenced from the ACCESS item, the value of the RESULTx item is written according to the convention of the ACCESS item.

(4) Reference to results

Write references to results (RESULTx) in items that apply the results of CONDITION part.

Results can only be referenced from elements of BLOCK_PARAMETER parts. They cannot be referenced from anything else.

In the example in Figure 5-39, the value of AvgOrSample and TimeOrCount of BLOCK_PARAMETER part are judged, and the values applied to the ACCESS item of AvgTime and AvgCount are determined. This proceeds as follows.

- If the value of AvgOrSample is 0 ⇒ ACCESS item of AvgTime is NA and ACCESS item of AvgCount is NA
- If the value of AvgOrSample is 1 and the value of TimeOrCount is 0 ⇒ ACCESS item of AvgTime is NA and ACCESS item of AvgCount is RW
- If the value of AvgOrSample is 1 and the value of TimeOrCount is 1 ⇒ ACCESS item of AvgTime is RW and ACCESS item of AvgCount is NA
- Any other situation ⇒ ACCESS item of AvgTime is NA and ACCESS item of AvgCount is NA

5.5 Parts and Elements with Defined Applications

In CSP+ specifications, parts other than the FILE_INFO, DEVICE_INFO, COMM_IF_INFO, and BLOCK_INFO parts do not specify elements that should be commonly included for all modules. However, when a module has certain functions or information, there are rules related to the elements used to express such functions or information. These rules exist as optional description specifications. This section describes the specifications of such elements.

The applications of the elements described in this section are predetermined, and thus the label names defined in this section cannot be used for any other application. Additionally, while the specifications of these optional descriptions indicate that descriptions of the functions or information described in this section are preferably included in accordance with the specifications described in this chapter when present, the creator of the CSP+ file is free to decide whether or not to include the descriptions. If the descriptions are included, however, they must be described in accordance with the specifications described herein. For the network availability for each application, see Table 5-304.

Table 5-304 Network Usage List for Each Application

Application	Network			
	CC-Link	CC-Link IEF	Ethernet*1	CC-Link IE TSN
Elements related to system area	Section 5.5.1	Section 5.5.1	-	Section 5.5.1
Parts and elements related to parameter initial settings based on initial settings	Section 5.5.2	Section 5.5.2	-	-
Parts and elements related to error history information acquisition	-	Section 5.5.3	-	-
METHOD specifications for conducting error checks of equipment	Section 5.5.4	Section 5.5.4	-	-
Parts and elements indicating parameters for safety communication function	-	Section 5.5.5	-	Section 5.5.10.3
MESSAGE specifications for writing the safety communication function parameter	-	Section 5.5.6	-	Section 5.5.10.4
Conventions of read/write capable parameters using parameter IDs	-	-	Section 5.5.7	-
Parts and elements related to energy management information acquisition	Section 5.5.8	Section 5.5.8	-	Section 5.5.8
Convention related to status code	-	-	Section 5.5.9	-
Parts and elements related to error related information	-	-	-	Section 5.5.10.1
Convention related to MESSAGE for slave station parameter automatic setting	-	-	-	Section 5.5.10.2

*: Will not be able to be used on the target network.

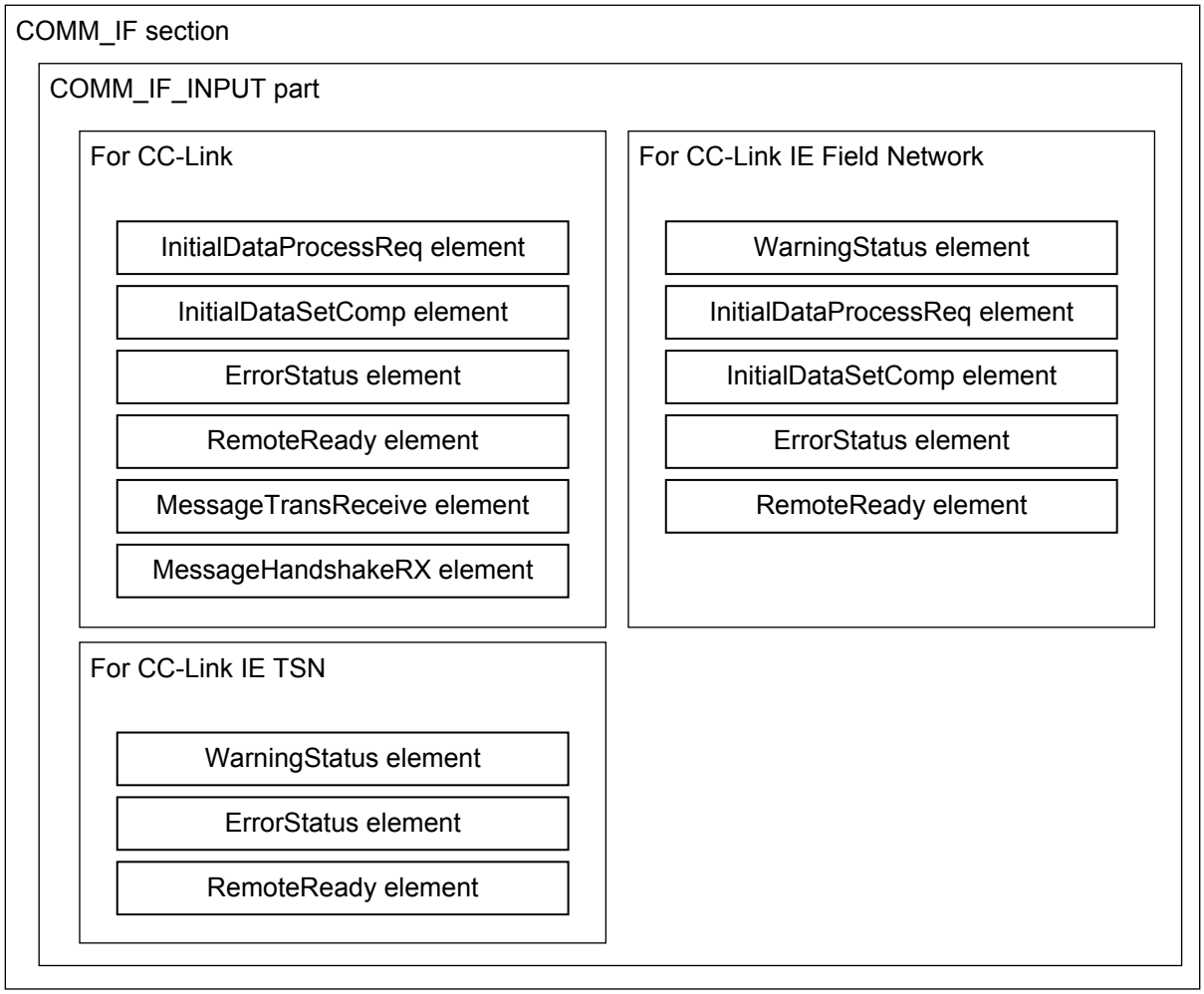
*1 For the Ethernet connected device, this indicates the CC-Link IE Field Network Basic communication and SLMP communication.

5.5.1 Elements related to system area

The specifications related to the system area is specified in CC-Link, CC-Link IE Field Network, and CC-Link IE TSN. This section describes the specifications of elements in a case where a flag of the system area is to be described.

5.5.1.1 Overview

A flag of the system area is described in the COMM_IF_INPUT or COMM_IF_OUTPUT part. Figure 5-40 shows images of the respective structures when a system area flag is described for a case where the communication interface is CC-Link, CC-Link IE Field Network, and CC-Link IE TSN.



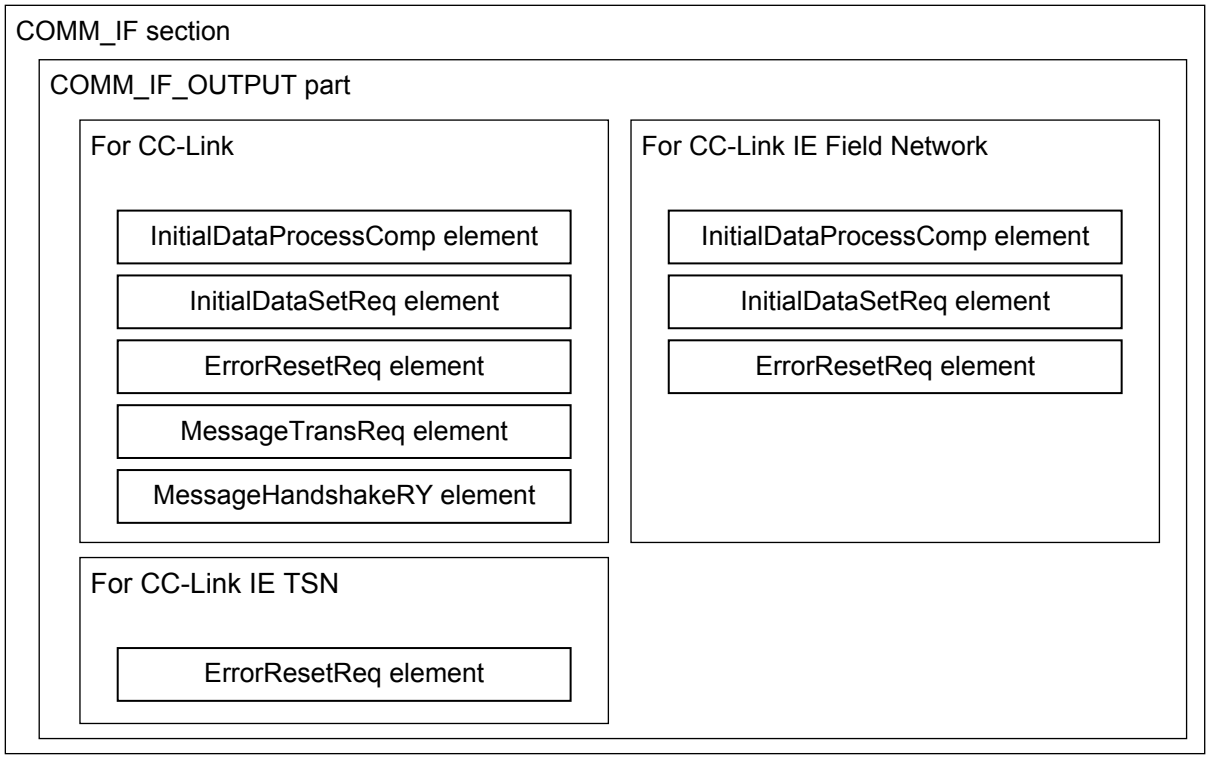


Figure 5-40 Image of Structure When System Area Is Described

Table 5-305, Table 5-306 and Table 5-307 list the elements described in the COMM_IF_INPUT part. For the detailed description specifications of each element, see Section 5.5.1.2.

Table 5-305 Element List When the Communication Interface Described in the COMM_IF_INPUT Part is CC-Link

No.	Element Name	Description Details	Required/Optional
1.	InitialDataProcessReq	Describes the initial data processing request flag.	Optional
2.	InitialDataSetComp	Describes the initial data setting completed flag.	Optional
3.	ErrorStatus	Describes the error status flag.	Optional
4.	RemoteReady	Describes the remote ready flag.	Optional
5.	MessageTransReceive	Describes message transmission reception.	Optional
6.	MessageHandshakeRX	Describes the input_message handshake flag.	Optional

Table 5-306 Element List When the Communication Interface Described in the COMM_IF_INPUT Part is the CC-Link IE Field Network

No.	Element Name	Description Details	Required/Optional
1.	WarningStatus	Describes the warning status flag.	Optional
2.	InitialDataProcessReq	Describes the initial data processing requirement flag.	Optional
3.	InitialDataSetComp	Describes the initial data setting completed flag.	Optional
4.	ErrorStatus	Describes the error status flag.	Optional
5.	RemoteReady	Describes the remote ready flag.	Optional

Table 5-307 Element List When the Communication Interface Described in the COMM_IF_INPUT Part is CC-Link IE TSN

No.	Element Name	Description Details	Required/Optional
1.	WarningStatus	Describes the warning status flag.	Optional
2.	ErrorStatus	Describes the error status flag.	Optional
3.	RemoteReady	Describes the remote ready flag.	Optional

Table 5-308, Table 5-309 and Table 5-310 list the elements described in the COMM_IF_OUTPUT part. For the detailed description specifications of each element, see Section 5.5.1.3.

Table 5-308 Element List When the Communication Interface Described in the COMM_IF_OUTPUT Part is CC-Link

No.	Element Name	Description Details	Required/Optional
1.	InitialDataProcessComp	Describes the initial data processing completed flag.	Optional
2.	InitialDataSetReq	Describes the initial data setting request flag.	Optional
3.	ErrorResetReq	Describes the error reset request flag.	Optional
4.	MessageTransReq	Describes message transmission request.	Optional
5.	MessageHandshakeRY	Describes the message handshake flag.	Optional

Table 5-309 Element List When the Communication Interface Described in the COMM_IF_OUTPUT Part is the CC-Link Field Network

No.	Element Name	Description Details	Required/Optional
1.	InitialDataProcessComp	Describes the initial data request completed flag.	Optional
2.	InitialDataSetReq	Describes the initial data setting request flag.	Optional
3.	ErrorResetReq	Describes the error reset request flag.	Optional

Table 5-310 Element List When the Communication Interface Described in the COMM_IF_OUTPUT Part is CC-Link IE TSN

No.	Element Name	Description Details	Required/Optional
1.	ErrorResetReq	Describes the error reset request flag.	Optional

5.5.1.2 Description details of elements described in COMM_IF_INPUT part

- **Description of system area flag elements when the communication interface is CC-Link**
  The following describes the description details of the elements described in the COMM_IF_INPUT part which are system area flag elements described when the communication interface is CC-Link.

(1) Item description of InitialDataProcessReq element

Table 5-311 Item Description Example of InitialDataProcessReq Element

No.	Item Name	Description Details
1.	LABEL	Write "InitialDataProcessReq".
2.	LABEL2	Write "InitialDataProcessReq". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Initial Data Processing Request Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote input address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.

(a) Item description example of InitialDataProcessReq element

```
<p:commIfInputMember label="InitialDataProcessReq">
   <p:label2><p:item>InitialDataProcessReq</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Initial Data Processing Request Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RX18</p:item></p:assign>
</p:commIfInputMember>
```

(2) Item description of InitialDataSetComp element

Table 5-312 Item Description Specifications of InitialDataSetComp Element

No.	Item Name	Description Details
1.	LABEL	Write "InitialDataSetComp".
2.	LABEL2	Write "InitialDataSetComp". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Initial Data Processing Complete". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote input address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.

(a) Item description example of InitialDataSetComp element

```
<p:commIfInputMember label="InitialDataSetComp">
   <p:label2><p:item>InitialDataSetComp</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Initial Data Processing Complete</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RX19</p:item></p:assign>
</p:commIfInputMember>
```

(3) Item description of ErrorStatus element

Table 5-313 Item Description Specifications of ErrorStatus Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrorStatus".
2.	LABEL2	Write "ErrorStatus". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Error Status Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote input address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.

(a) Item description example of ErrorStatus element

```
<p:commIfInputMember label="ErrorStatus">
   <p:label2><p:item>ErrorStatus</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Error Status Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RX1A</p:item></p:assign>
</p:commIfInputMember>
```

(4) Item description of RemoteReady element

Table 5-314 Item Description Specifications of RemoteReady Element

No.	Item Name	Description Details
1.	LABEL	Write "RemoteReady".
2.	LABEL2	Write "RemoteReady". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Remote Ready". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote input address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.

(a) Item description example of RemoteReady element

```
<p:commIfInputMember label="RemoteReady">
   <p:label2><p:item>RemoteReady</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Remote Ready</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RX1B</p:item></p:assign>
</p:commIfInputMember>
```

(5) Item description of MessageTransReceive element

Table 5-315 Item Description Specifications of MessageTransReceive Element

No.	Item Name	Description Details
1.	LABEL	Write "MessageTransReceive".
2.	LABEL2	Write "MessageTransReceive". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Message Transmission Receive". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote input address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.

(a) Item description example of MessageTransReceive element

```
<p:commIfInputMember label="MessageTransReceive">
   <p:label2><p:item>MessageTransReceive</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Message Transmission Receive</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RX1C</p:item></p:assign>
</p:commIfInputMember>
```

(6) Item description of MessageHandshakeRX element

Table 5-316 Item Description Specifications of MessageHandshakeRX Element

No.	Item Name	Description Details
1.	LABEL	Write "MessageHandshakeRX".
2.	LABEL2	Write "MessageHandshakeRX". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Input_Message Handshake Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote input address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.

(a) Item description example of MessageHandshakeRX element

```
<p:commIfInputMember label="MessageHandshakeRX">
   <p:label2><p:item>MessageHandshakeRX</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Input_Message Handshake Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RX1D</p:item></p:assign>
</p:commIfInputMember>
```

- **Description of system area flag elements when the communication interface is the CC-Link IE Field Network**
  The following describes the description details of the elements described in the COMM_IF_INPUT part which are system area flag elements described when the communication interface is the CC-Link IE Field Network.

(7) Item description of WarningStatus element

Table 5-317 Item Description Specifications of WarningStatus Element

No.	Item Name	Description Details
1.	LABEL	Write "WarningStatus".
2.	LABEL2	Write "WarningStatus". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Warning Status Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote input address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.

(a) Item description example of WarningStatus element

```
<p:commIfInputMember label="WarningStatus">
   <p:label2><p:item>WarningStatus</p:item></p:label2>
   <p:category><p:item> System area </p:item></p:category>
   <p:name><p:item>Warning Status Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RX18</p:item></p:assign>
</p:commIfInputMember>
```

(8) Item description of InitialDataProcessReq element

Table 5-318 Item Description Example of InitialDataProcessingReq Element

No.	Item Name	Description Details
1.	LABEL	Write "InitialDataProcessReq".
2.	LABEL2	Write "InitialDataProcessReq". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Initial Data Processing Request Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote input address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.

(a) Item description example of InitialDataProcessReq element

```
<p:commIfInputMember label="InitialDataProcessReq">
   <p:label2><p:item>InitialDataProcessReq</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Initial Data Processing Request Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RX18</p:item></p:assign>
</p:commIfInputMember>
```

(9) Item description of InitialDataSetComp element

Table 5-319 Item Description Specifications of InitialDataSetComp Element

No.	Item Name	Description Details
1.	LABEL	Write "InitialDataSetComp".
2.	LABEL2	Write "InitialDataSetComp". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Initial Data Setting Complete Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote input address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.

(a) Item description example of InitialDataSetComp element

```
<p:commIfInputMember label="InitialDataSetComp">
   <p:label2><p:item>InitialDataSetComp</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Initial Data Setting Complete Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RX19</p:item></p:assign>
</p:commIfInputMember>
```

(10) Item description of ErrorStatus element

Table 5-320 Item Description Specifications of ErrorStatus Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrorStatus".
2.	LABEL2	Write "ErrorStatus". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Error Status Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote input address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.

(a) Item description example of ErrorStatus element

```
<p:commIfInputMember label="ErrorStatus">
   <p:label2><p:item>ErrorStatus</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Error Status Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RX1A</p:item></p:assign>
</p:commIfInputMember>
```

(11) Item description of RemoteReady element

Table 5-321 Item Description Specifications of RemoteReady Element

No.	Item Name	Description Details
1.	LABEL	Write "RemoteReady"
2.	LABEL2	Write "RemoteReady". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "RemoteReady". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote input address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.

(a) Item description example of RemoteReady element

```
<p:commIfInputMember label="RemoteReady">
   <p:label2><p:item>RemoteReady</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Remote Ready</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RX1B</p:item></p:assign>
</p:commIfInputMember>
```

- **Description of system area flag elements when the communication interface is CC-Link IE TSN**
  The following describes the description details of the elements described in the COMM_IF_INPUT part which are system area flag elements described when the communication interface is CC-Link IE TSN.

(12) Item description of WarningStatus element

Table 5-322 Item Description Example of WarningStatus Element

No.	Item Name	Description Details
1.	LABEL	Write "WarningStatus".
2.	LABEL2	Write "WarningStatus". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Warning Status Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote input address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.

(a) Item description example of WarningStatus element

```
<p:commIfInputMember label="WarningStatus">
   <p:label2><p:item>WarningStatus</p:item></p:label2>
   <p:category><p:item>System Area</p:item></p:category>
   <p:name><p:item>Warning Status Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RX18</p:item></p:assign>
</p:commIfInputMember>
```

(13) Item description of ErrorStatus element

Table 5-323 Item Description Specifications of ErrorStatus Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrorStatus".
2.	LABEL2	Write "ErrorStatus". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Error Status Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote input address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.

(a) Item description example of ErrorStatus element

```
<p:commIfInputMember label="ErrorStatus">
   <p:label2><p:item>ErrorStatus</p:item></p:label2>
   <p:category><p:item>System Area</p:item></p:category>
   <p:name><p:item>Error Status Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RX1A</p:item></p:assign>
</p:commIfInputMember>
```

(14) Item description of RemoteReady element

Table 5-324 Item Description Specifications of RemoteReady Element

No.	Item Name	Description Details
1.	LABEL	Write "RemoteReady".
2.	LABEL2	Write "RemoteReady". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Remote Ready". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote input address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_INPUT part.

(a) Item description example of RemoteReady element

```
<p:commIfInputMember label="RemoteReady">
   <p:label2><p:item>RemoteReady</p:item></p:label2>
   <p:category><p:item>System Area</p:item></p:category>
   <p:name><p:item>Remote Ready</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RX1B</p:item></p:assign>
</p:commIfInputMember>
```

5.5.1.3 Description details of elements described in COMM_IF_OUTPUT part

- **Description of system area flag elements when the communication interface is CC-Link**
  The following describes the description details of the elements described in the COMM_IF_OUTPUT part which are system area flag elements described when the communication interface is CC-Link.

(1) Item description of InitialDataProcessComp element

Table 5-325 Item Description Example of InitialDataProcessComp Element

No.	Item Name	Description Details
1.	LABEL	Write "InitialDataProcessComp".
2.	LABEL2	Write "InitialDataProcessComp". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Initial Data Processing Complete Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote output address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.

(a) Item description example of InitialDataProcessComp element

```
<p:commIfOutputMember label="InitialDataProcessComp">
   <p:label2><p:item>InitialDataProcessComp</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Initial Data Processing Complete Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RY18</p:item></p:assign>
</p:commIfOutputMember>
```

(2) Item description of InitialDataSetReq element

Table 5-326 Item Description Specifications of InitialDataSetReq Element

No.	Item Name	Description Details
1.	LABEL	Write "InitialDataSetReq".
2.	LABEL2	Write "InitialDataSetReq". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Initial Data Setting Request Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote output address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.

(a) Item description example of InitialDataSetReq element

```
<p:commIfOutputMember label="InitialDataSetReq">
   <p:label2><p:item>InitialDataSetReq</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Initial Data Setting Request Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RY19</p:item></p:assign>
</p:commIfOutputMember>
```

(3) Item description of ErrorResetReq element

Table 5-327 Item Description Specifications of ErrorResetReq Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrorResetReq".
2.	LABEL2	Write "ErrorResetReq". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Error Reset Request Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote output address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.

(a) Item description example of ErrorResetReq element

```
<p:commIfOutputMember label="ErrorResetReq">
   <p:label2><p:item>ErrorResetReq</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Error Reset Request Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RY1A</p:item></p:assign>
</p:commIfOutputMember>
```

(4) Item description of MessageTransReq element

Table 5-328 Item Description Specifications of MessageTransReq Element

No.	Item Name	Description Details
1.	LABEL	Write "MessageTransReq".
2.	LABEL2	Write "MessageTransReq". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Message Transmission Request". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote output address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.

(a) Item description example of MessageTransReq element

```
<p:commIfOutputMember label="MessageTransReq">
   <p:label2><p:item>MessageTransReq</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Message Transmission Request</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RY1C</p:item></p:assign>
</p:commIfOutputMember>
```

(5) Item description of MessageHandshakeRY element

Table 5-329 Item Description Specifications of MessageHandshakeRY Element

No.	Item Name	Description Details
1.	LABEL	Write "MessageHandshakeRY".
2.	LABEL2	Write "MessageHandshakeRY". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Output_Message Handshake Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote output address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.

(a) Item description example of MessageHandshakeRY element

```
<p:commIfOutputMember label="MessageHandshakeRY">
   <p:label2><p:item>MessageHandshakeRY</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Output_Message Handshake Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RY1D</p:item></p:assign>
</p:commIfOutputMember>
```

- **Description of system area flag elements when the communication interface is the CC-Link IE Field Network**
  The following describes the description details of the elements described in the COMM_IF_OUTPUT part which are system area flag elements described when the communication interface is the CC-Link IE Field Network.

- **Description of system area flag elements when the communication interface is CC-Link Field Network**
  The following describes the description details of the elements described in the COMM_IF_OUTPUT part which are system area flag elements described when the communication interface is CC-Link Field Network.

(6) Item description of InitialDataProcessComp element

Table 5-330 Item Description Example of InitialDataProcessComp Element

No.	Item Name	Description Details
1.	LABEL	Write "InitialDataProcessComp".
2.	LABEL2	Write "InitialDataProcessComp". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Initial Data Processing Complete Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote output address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.

(a) Item description example of InitialDataProcessComp element

```
<p:commIfOutputMember label="InitialDataProcessComp">
   <p:label2><p:item>InitialDataProcessComp</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Initial Data Processing Complete Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RY18</p:item></p:assign>
</p:commIfOutputMember>
```

(7) Item description of InitialDataSetReq element

Table 5-331 Item Description Specifications of InitialDataSetReq Element

No.	Item Name	Description Details
1.	LABEL	Write "InitialDataSetReq".
2.	LABEL2	Write "InitialDataSetReq". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Initial Data Setting Request Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote output address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.

(a) Item description example of InitialDataSetReq element

```
<p:commIfOutputMember label="InitialDataSetReq">
   <p:label2><p:item>InitialDataSetReq</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Initial Data Setting Request Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RY19</p:item></p:assign>
</p:commIfOutputMember>
```

(8) Item description of ErrorResetReq element

Table 5-332 Item Description Specifications of ErrorResetReq Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrorResetReq".
2.	LABEL2	Write "ErrorResetReq". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Error Reset Request Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote output address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.

(a) Item description example of ErrorResetReq element

```
<p:commIfOutputMember label="ErrorResetReq">
   <p:label2><p:item>ErrorResetReq</p:item></p:label2>
   <p:category><p:item>System area</p:item></p:category>
   <p:name><p:item>Error Reset Request Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RY1A</p:item></p:assign>
</p:commIfOutputMember>
```

- **Description of system area flag elements when the communication interface is CC-Link IE TSN**
  The following describes the description details of the elements described in the COMM_IF_OUTPUT part which are system area flag elements described when the communication interface is CC-Link IE TSN.

(9) Item description of ErrorResetReq element

Table 5-333 Item Description Specifications of ErrorResetReq Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrorResetReq".
2.	LABEL2	Write "ErrorResetReq". [Recommended specifications]
3.	CATEGORY	Write "System area". [Recommended specifications]
4.	NAME	Write "Error Reset Request Flag". [Recommended specifications]
5.	DATATYPE	Write "BOOL".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write prohibited.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RF".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Note that this item is normally not omitted, and the written value is the corresponding remote output address.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.
13.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part. Normally not written.
14.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_OUTPUT part.

(a) Item description example of ErrorResetReq element

```
<p:commIfOutputMember label="ErrorResetReq">
   <p:label2><p:item>ErrorResetReq</p:item></p:label2>
   <p:category><p:item>System Area</p:item></p:category>
   <p:name><p:item>Error Reset Request Flag</p:item></p:name>
   <p:datatype><p:item>BOOL</p:item></p:datatype>
   <p:access><p:item>RF</p:item></p:access>
   <p:assign><p:item>RY1A</p:item></p:assign>
</p:commIfOutputMember>
```

5.5.2 Parts and elements related to parameter initialization based on initial settings

CC-Link and the CC-Link IE Field Network define specifications related to parameter initialization based on initial settings. This section describes the specifications of parts and elements in a case where parameter initialization based on initial settings is to be described.

5.5.2.1 Overview

Parameter initialization based on initial settings is described in the METHOD part and COMM_IF_PARAMETER part. Figure 5-41 shows an image of the structure when parameter initialization based on initial settings is described. (These specifications are common to both CC-Link and CC-Link IE Field Network communication interfaces.)

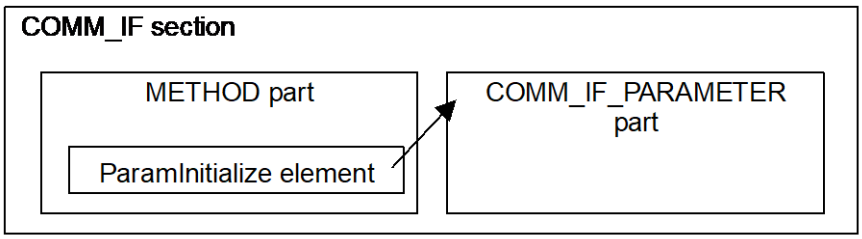


Figure 5-41 Image of Structure When Parameter Initialization Based on Initial Settings Is Described

Table 5-334 shows a list of elements described in the METHOD part. For details of the description specifications of the elements, see Section 5.5.2.2.

Table 5-334 List of Elements Described in METHOD Part

No.	Element Name	Description Details	Required/Optional
1.	ParamInitialize	Describes the method used to initialize parameters based on initial settings.	Required

Table 5-335 lists the elements described in the COMM_IF_PARAMETER part. For details of the description specifications of the elements, see Section 5.5.2.3.

Table 5-335 List of Elements Described in COMM_IF_PARAMETER Part

No.	Element Name	Description Details	Required/Optional
1.	(Any name)	Describes the parameter information to be set by parameter initialization based on initial settings. Requires description of all parameters subject to initialization based on initial settings as elements.	Required

While multiple parts are used to describe parameter initialization based on initial settings, there are no regulations related to the label name of each part. (The CSP+ file creator can freely assign the label names.)

Further, there are no regulations related to the method used to change parameters set by initial settings. Write the changes in accordance with the description specifications of the METHOD part.

5.5.2.2 Description details of elements described in METHOD part

(1) Item description of ParamInitialize element

Table 5-336 Item Description Specifications of ParamInitialize Element

No.	Item Name	Description Details
1.	LABEL	Write "ParamInitialize".
2.	LABEL2	Write "ParamInitialize". [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular METHOD part.
4.	NAME	Write "Initial Setting". [Recommended specifications]
5.	TARGET	Write "'Part name'.* ". For 'Part name', write the label name of the COMM_IF_PARAMETER part where the parameters specified in Table 5-335 are described.
6.	METHOD_TYPE	Write "OTHER".
7.	WRITE_REGISTER	Write "<$(*.ASSIGN)>".
8.	WRITE_DATA	Write "<$(*.VALLUE)>".
9.	WRITE_DATATYPE	Write "<$(*.DATATYPE)>".
10.	READ_REGISTER	Write prohibited.
11.	READ_DATA	Write prohibited.
12.	READ_DATATYPE	Write prohibited.
13.	INTERLOCK	Write that the initial data processing request flag is ON. The description specifications are the same as those for a regular METHOD part.
14.	REQ_FLAG	Write that the initial data processing completed flag is to be turned ON, and the initial data setting request flag is to be turned ON, in that order. The descriptions must be written in this order. The description specifications are the same as those for a regular METHOD part.
15.	END_CONDITION	Write that the initial data processing request flag is OFF, and the initial data setting completed flag is ON, in that order. The descriptions must be written in this order. The description specifications are the same as those for a regular METHOD part.
16.	ERR_CONDITION	Not specified. Write a value in the same manner as the regular METHOD part.
17.	ERR_REGISTER	Not specified. Write a value in the same manner as the regular METHOD part.
18.	ERR_CODE_RANGE	Not specified. Write a value in the same manner as the regular METHOD part.
19.	RELATED_METHOD	Write prohibited.
20.	COMMENT	Not specified. Write a value in the same manner as the regular METHOD part.

(a) Item description example of ParamInitialize element

```
<p:methodMember label="ParamInitialize">
   <p:label2><p:item>ParamInitialize</p:item></p:label2>
   <p:name><p:item>Initial Setting</p:item></p:name>
   <p:target><p:item>PrmTable.*</p:item></p:target>
   <p:methodType><p:item>OTHER</p:item></p:methodType>
   <p:writeRegister><p:item>&lt;$(*.ASSIGN)&gt;</p:item></p:writeRegister>
   <p:writeData><p:item>&lt;$(*.VALUE)&gt;</p:item></p:writeData>
   <p:writeDatatype><p:item>&lt;$(*.DATATYPE)&gt;</p:item></p:writeDatatype>
   <p:interlock><p:item>RX18==ON</p:item></p:interlock>
   <p:reqFlag><p:item>RY18=ON</item><item>RY19=ON</p:item></p:reqFlag>
   <p:endCondition><p:item>RX18==OFF&amp;amp;&amp;amp;RX19==ON</p:item></p:endCondition>
   <p:comment><p:item>Batch Parameter Setting with Initial Setting</p:item></p:comment>
</p:methodMember>
```

5.5.2.3 Description details of elements described in COMM_IF_PARAMETER part

(1) Item description of element (of any name)

Table 5-337 Item Description Specifications of Parameter1 Element

No.	Item Name	Description Details
1.	LABEL	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
4.	NAME	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
5.	DATATYPE	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
11.	WRITE_ORDER	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
12.	ASSIGN	Write cyclic address information for setting parameters based on initial settings. The description specifications are the same as those for a regular COMM_IF_PARAMETER part.
13.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
14.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
15.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.

(a) Item description example of Parameter1 element

```
<p:commIfParameterMember label="Parameter1">
   <p:label2><p:item>Parameter1</p:item></p:label2>
   <p:name><p:item>Parameter 1</p:item></p:name>
   <p:access><p:item>RW</p:item></p:access>
   <p:assign><p:item>RWw9</p:item></p:assign>
   <p:ref><p:item>BlockSec.BlockParameter.Parameter1</p:item></p:ref>
</p:commIfParameterMember>
```

5.5.3 Parts and elements related to acquisition of error logs

CC-Link IE Field Network defines specifications related to error history information acquisition.

The module has a function that records the information of any errors that occur during module operation, and acquires that error information as logs via SLMP via an external source. This section describes the specifications of parts and elements in a case where error log acquisition processing is to be described.

5.5.3.1 Overview

Error log acquisition is described in the MESSAGE, COMM_IF_COMMAND, BLOCK_COMMAND, and COMMAND_ARGUMENT parts. Error logs are sometimes managed on the module main body side and sometimes on the communication module side, depending on the module mode and specifications. The parts that are described, therefore, differ accordingly (parts can be described based on the concept of the COMM_IF section and BLOCK section). Figure 5-42 and Figure 5-43 show images of the structure when error log acquisition is described. (The specifications are for CC-Link IE Field Network since a MESSAGE part is used.)

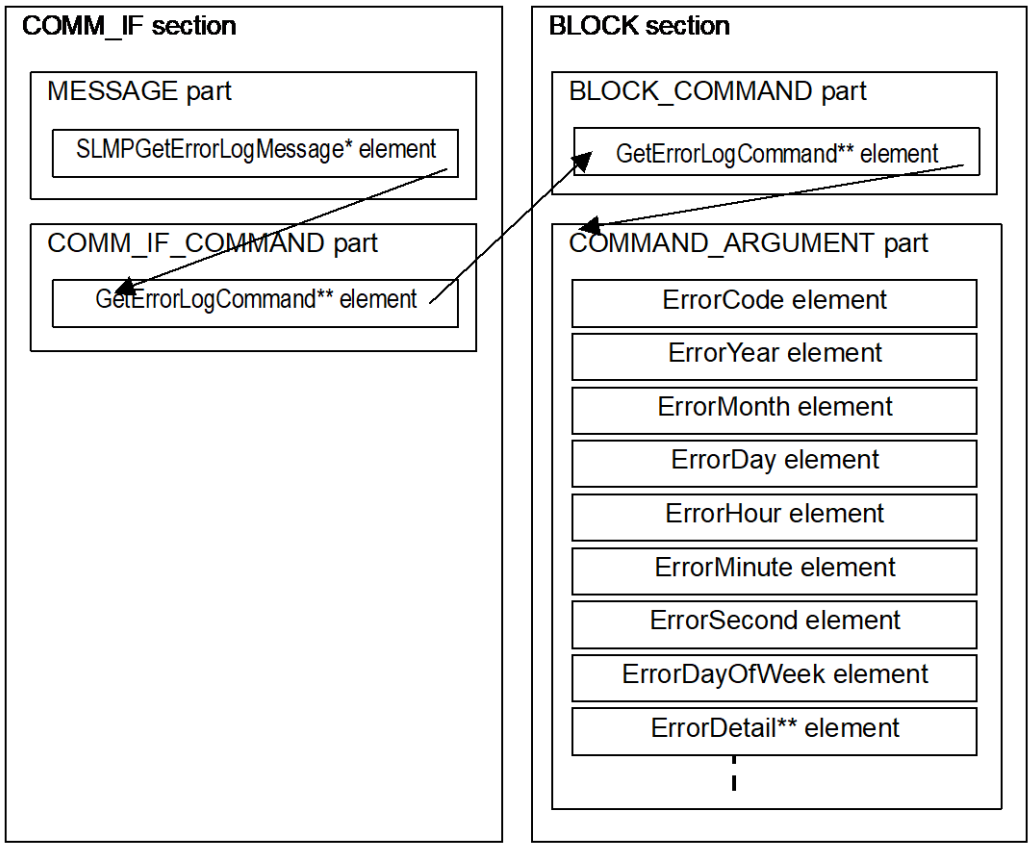


Figure 5-42 Image of Structure When Error Log Acquisition Is Described (When Error Logs Are Managed on Module Main Body Side)

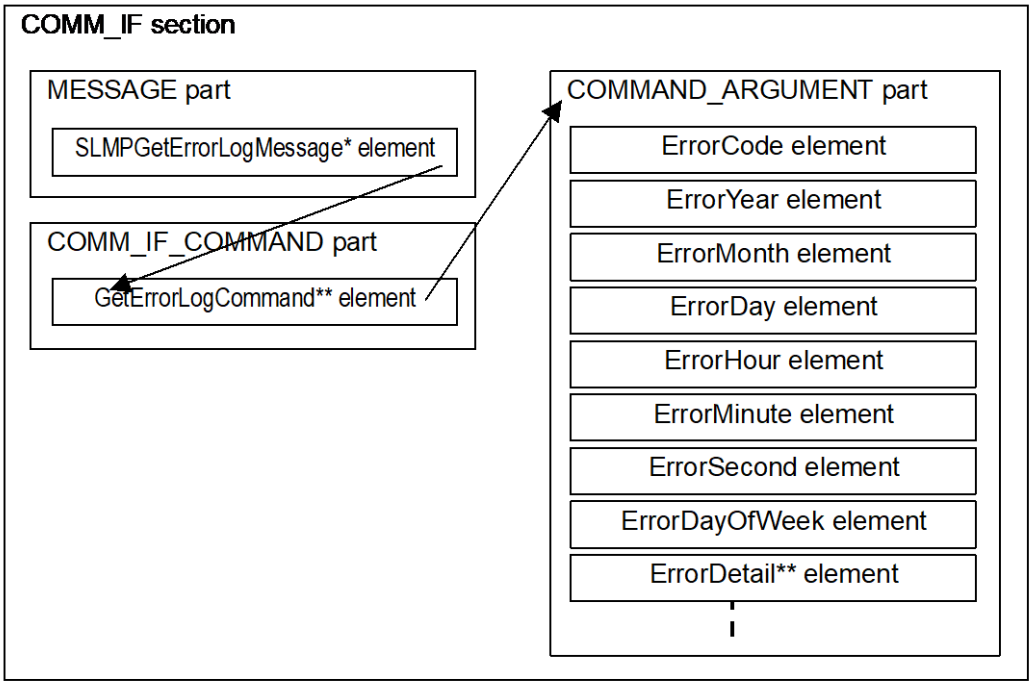


Figure 5-43 Image of Structure When Error Log Acquisition Is Described (When Error Logs Are Managed on Communication Module Side)

Table 5-338 shows a list of elements described in the MESSAGE part. For details of the description specifications of the elements, see Section 5.5.3.2.

Table 5-338 List of Elements Described in MESSAGE Part

No.	Element Name	Description Details	Required/Optional
1.	SLMPGetErrorLogMessage*	Describes the message used to acquire error logs. For "*", see detailed description specifications.	Required

Table 5-339 lists the elements described in the COMM_IF_COMMAND part. For details of the description specifications of the elements, see Section 5.5.3.3.

Table 5-339 List of Elements Described in COMM_IF_COMMAND Part

No.	Element Name	Description Details	Required/Optional
1.	GetErrorLogCommand**	Describes the command information to be used for error log acquisition. For "**", see detailed description specifications.	Required

Table 5-340 lists the elements described in the BLOCK_COMMAND part. For details of the description specifications of the elements, see Section 5.5.3.4.

Table 5-340 List of Elements Described in BLOCK_COMMAND Part

No.	Element Name	Description Details	Required/Optional
1.	GetErrorLogCommand**	Describes the command information to be used for error log acquisition. For "**", see detailed description specifications.	Required*1

*1. Required only when a BLOCK_COMMAND part is used. Optional when a BLOCK_COMMAND part is not used.

Table 5-341 lists the elements described in the COMMAND_ARGUMENT part (specifications common to COMM_IF and BLOCK sections). While all elements are optionally described, at least one element must be described. (A description of only the COMMAND_ARGUMENT part in which these elements are not described is not permitted.) For details of the description specifications of the elements, see Section 5.5.3.5.

Table 5-341 List of Elements Described in COMMAND_ARGUMENT Part

No.	Element Name	Description Details	Required/Optional
1.	ErrorCode	Describes the error code.	Optional
2.	ErrorYear	Describes the error occurrence date/time (year). Recommended: Write a description that can be read as the four digits of the western calendar.	Optional (However, omission of elements between two included elements is prohibited.*1)
3.	ErrorMonth	Describes the error occurrence date/time (month).	
4.	ErrorDay	Describes the error occurrence date/time (day).	
5.	ErrorHour	Describes the error occurrence date/time (hour).	
6.	ErrorMinute	Describes the error occurrence date/time (minute).	
7.	ErrorSecond	Describes the error occurrence date/time (second).	
8.	ErrorDayOfWeek	Describes the error occurrence date/time (day of week).	Optional
9.	ErrorDetail**	Describes the error log details. For "**", see detailed description specifications. Note that multiple details can be described by changing the information following the prefix. (Example: Frequency at error occurrence, output value at error occurrence)	Optional
10.	(Any name)	Application not specified. May be used for the following applications: Example: To describe dummy elements and express the data order when error logs are acquired, including unnecessary data between error log details 1 and error log details 2. (With a message, the order of the elements of the COMMAND_ARGUMENT part s to correspond with the order of the data during data reception.)	Optional

*1. For example, while it is possible to simultaneously omit ErrorSecond and ErrorMinute, you cannot omit ErrorMinute in a case where both ErrorHour and ErrorSecond are described. With the utility software, we recommend that you do not display omitted date/time information (non-acquirable date/time information), but rather display only the date/time information described in the COMMAND_ARGUMENT part.

Note that while multiple parts are used to describe error log acquisition, there are no regulations regarding the label name of each part. (The CSP+ file creator can freely assign label names.)

5.5.3.2 Description details of elements described in MESSAGE part

(1) Item description of SLMPGetErrorLogMessage*

Table 5-342 Item Description Specifications for SLMPGetErrorLogMessage* Element

No.	Item Name	Description Details
1.	LABEL	Write "SLMPGetErrorLogMessage*". Write a sequential numerical value in the area marked by an asterisk (*), starting from 1. (99, maximum) (Example: SLMPGetErrorLogMessage1, SLMPGetErrorLogMessage2, …)
2.	LABEL2	Write "SLMPGetErrorLogMessage*". [Recommended specifications] Write the same value as that of the LABEL item in the area marked by an asterisk (*).
3.	CATEGORY	Not specified. Write a value in the same manner as the regular MESSAGE part.
4.	NAME	Write "Error History Acquisition*". [Recommended specifications] Write the same value as that of the LABEL item in the area marked by an asterisk (*).
5.	TARGET	Write a reference to the COMM_IF_COMMAND part specified by Section 5.5.3.3.
6.	MESSAGE_TYPE	Write "COMMAND". However, in a case where a message that acquires multiple error history information logs using "SEQ" in RELATED_MESSAGE is consecutively executed, you can also write "OTHER" in MESSAGE TYPE for the MESSAGE element that serves as the reference destination by SEQ. At this time, write "COMMAND" in MESSAGETYPE for the MESSAGE element that serves as the reference source.
7.	REQUEST_TYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
8.	REQUEST_DATA	Not specified. Write a value in the same manner as the regular MESSAGE part.
9.	REQUEST_DATATYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
10.	RESPONSE_TYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
11.	RESPONSE_DATA	Write a reference to the VALUE of an element of the COMMAND_ARGUMENT part specified in Table 5-341. A minimum of one reference description to the VALUE of the element is required.
12.	RESPONSE_DATATYPE	Write the data type (representation string) or a reference to the DATATYPE item of an element of the COMMAND_ARGUMENT part specified in Table 5-341.
13.	ERR_TYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
14.	ERR_CODE_RANGE	Not specified. Write a value in the same manner as the regular MESSAGE part.
15.	RELATED_MESSAGE	Not specified. Write a value in the same manner as the regular MESSAGE part. You can also describe consecutive execution of a message that acquires multiple error history information logs using the keyword "SEQ". In such a case, writing a message of a label other than "SLMPGetErrorLogMessage*" in the element referenced by SEQ is prohibited.
16.	COMMENT	Not specified. Write a value in the same manner as the regular MESSAGE part. Write an explanation of the error logs to be read if such an explanation is required. [Example: Write that only the lower two digits of the error occurrence date/time (year) are displayed if only the lower two digits can be acquired.]

(a) Item description example of SLMPGetErrorLogMessage1 element

```
<p:messageMember label="SLMPGetErrorLogMessage1">
   <p:label2><p:item>SLMPGetErrorLogMessage1</p:item></p:label2>
   <p:name><p:item>Error History Acquisition 1</p:item></p:name>
   <p:target><p:item>GetErrorLogCommand</p:item></p:target>
   <p:messageType><p:item>COMMAND</p:item></p:messageType>
   <p:requestType><p:item>rdReqST_Binary</p:item></p:requestType>
   <p:requestData>
      <p:item>&lt;0x0401&gt;&lt;0x0000&gt;&lt;0x00000000&gt;&lt;$0x0006&gt;</p:item>
   </p:requestData>
   <p:requestDatatype>
      <p:item>&lt;WORD&gt;&lt;WORD&gt;&lt;DWORD&gt;&lt;WORD&gt;</p:item>
   </p:requestDatatype>
   <p:responseType><p:item>rdResST_Binary</p:item></p:responseType>
   <p:responseData><p:item>$(ARGUMENT.*.VALUE)</p:item></p:responseData>
   <p:responseDatatype><p:item>$(ARGUMENT.*.DATATYPE)</p:item></p:responseDatatype>
   <p:errType><p:item>rdErr_Binary</p:item></p:errType>
   <p:errCodeRange><p:item>EnumErrorCode</p:item></p:errCodeRange>
</p:messageMember>
```

5.5.3.3 Description details of elements described in COMM_IF_COMMAND part

(1) Item description of GetErrorLogCommand** element

Table 5-343 Item Description of GetErrorLogCommand Element**

No.	Item Name	Description Details
1.	LABEL	Write "GetErrorLogCommand**". Write any string, including spaces, in the area marked by the asterisks (**).
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.
4.	NAME	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.
5.	ARGUMENT	[When error logs are managed on module main body side] Write prohibited. [When error logs are managed on communication module side] Write a reference to the COMMAND_ARGUMENT part where an element defined in Table 5-341 is described.
6.	REF	[When error logs are managed on module main body side] Write a reference to the BLOCK_COMMAND specified in Section 5.5.3.4. [When error logs are managed on communication module side] Write prohibited.
7.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.

(a) Item description example of GetErrorLogCommand element

```
<p:commIfCommandMember label="GetErrorLogCommand">
   <p:label2><p:item>GetErrorLogCommand</p:item></p:label2>
   <p:name><p:item>Error History Acquisition</p:item></p:name>
   <p:ref><p:item>BlockSec.BlockCommand.GetErrorLogCommand</p:item></p:ref>
</p:commIfCommandMember>
```

5.5.3.4 Description details of elements described in BLOCK_COMMAND part

(1) Item description of GetErrorLogCommand** element

Table 5-344 Item Description Specifications of GetErrorLogCommand Element**

No.	Item Name	Description Details
1.	LABEL	Write "GetErrorLogCommand**". Write any string, including spaces, in the area marked by the asterisks (**).
2.	LABEL2	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.
4.	NAME	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.
5.	ARGUMENT	Write a reference to the COMMAND_ARGUMENT part where an element specified in Table 5-341 is described.
6.	COMMENT	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.

(a) Item description example of GetErrorLogCommand element

```
<p:blockCommandMember label="GetErrorLogCommand">
   <p:name><p:item>GetErrorLogCommand</p:item></p:name>
   <p:argument><p:item>ErrorLogArgument</p:item></p:argument>
</p:blockCommandMember>
```

5.5.3.5 Description details of elements described in COMMAND_ARGUMENT part

(1) Item description of ErrorCode element

Table 5-345 Item Description Specifications of ErrorCode Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrorCode".
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
5.	DATATYPE	Select and write one of the following: bit string type, signed integer type, unsigned integer type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Note that you can define the meaning of an error code or comment by writing a reference to an ENUM part in this item.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of ErrorCode element

```
<p:commandArgumentMember label="ErrorCode">
   <p:label2><p:item>ErrorCode</p:item></p:label2>
   <p:name><p:item>Error Code</p:item></p:name>
   <p:datatype><p:item>WORD</p:item></p:datatype>
   <p:range><p:enumRefItem>EnumErrorCode</p:enumRefItem></p:range>
   <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(2) Item description of ErrorYear element

Table 5-346 Item Description Specifications of ErrorYear Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrorYear".
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
5.	DATATYPE	Select and write one of the following: unsigned integer type, BCD integer type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of ErrorYear element

```
<p:commandArgumentMember label="ErrorYear">
   <p:label2><p:item>ErrorYear</p:item></p:label2>
   <p:name><p:item>Error Occurrence Date and Time (Year)</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(3) Item description of ErrorMonth element

Table 5-347 Item Description Specifications of ErrorMonth Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrorMonth".
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
5.	DATATYPE	Select and write one of the following: unsigned integer type, BCD integer type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of ErrorMonth element

```
<p:commandArgumentMember label="ErrorMonth">
   <p:label2><p:item>ErrorMonth</p:item></p:label2>
   <p:name><p:item>Error Occurrence Date and Time (Month)</p:item></p:name>
   <p:datatype><p:item>UINT4</p:item></p:datatype>
   <p:range><p:item>[1,12]</p:item></p:range>
   <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(4) Item description of ErrorDay element

Table 5-348 Item Description Specifications of ErrorDay Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrorDay".
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
5.	DATATYPE	Select and write one of the following: unsigned integer type, BCD integer type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of ErrorDay element

```
<p:commandArgumentMember label="ErrorDay">
   <p:label2><p:item>ErrorDay</p:item></p:label2>
   <p:name><p:item>Error Occurrence Date and Time (Date)</p:item></p:name>
   <p:datatype><p:item>UINT8</p:item></p:datatype>
   <p:range><p:item>[1,31]</p:item></p:range>
   <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(5) Item description of ErrorHour element

Table 5-349 Item Description Specifications of ErrorHour Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrorHour".
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
5.	DATATYPE	Select and write one of the following: unsigned integer type, BCD integer type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of ErrorHour element

```
<p:commandArgumentMember label="ErrorHour">
   <p:label2><p:item>ErrorHour</p:item></p:label2>
   <p:name><p:item>Error Occurrence Date and Time (Hour)</p:item></p:name>
   <p:datatype><p:item>UINT8</p:item></p:datatype>
   <p:range><p:item>[0,23]</p:item></p:range>
   <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(6) Item description of ErrorMinite element

Table 5-350 Item Description Specifications of ErrorMinite Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrorMinite".
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
5.	DATATYPE	Select and write one of the following: unsigned integer type, BCD integer type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of ErrorMinite element

```
<p:commandArgumentMember label="ErrorMinite">
   <p:label2><p:item>ErrorMinite")</p:item></p:label2>
   <p:name><p:item>Error Occurrence Date and Time (Minute)</p:item></p:name>
   <p:datatype><p:item>UINT8</p:item></p:datatype>
   <p:range><p:item>[0,59]</p:item></p:range>
   <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(7) Item description of ErrorSecond element

Table 5-351 Item Description Specifications of ErrorSecond Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrorSecond".
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
5.	DATATYPE	Select and write one of the following: unsigned integer type, BCD integer type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of ErrorSecond element

```
<p:commandArgumentMember label="ErrorSecond">
   <p:label2><p:item>ErrorSecond</p:item></p:label2>
   <p:name><p:item>Error Occurrence Date and Time (Second)</p:item></p:name>
   <p:datatype><p:item>UINT8</p:item></p:datatype>
   <p:range><p:item>[0,59]</p:item></p:range>
   <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(8) Item description of ErrorDayOfWeek element

Table 5-352 Item Description Specifications of ErrorDayOfWeek Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrorDayOfWeek".
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
5.	DATATYPE	Select and write one of the following: unsigned integer type, BCD integer type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of ErrorDayOfWeek element

```
<p:commandArgumentMember label="ErrorDayOfWeek">
   <p:label2><p:item>ErrorDayOfWeek</p:item></p:label2>
   <p:name><p:item>Error Occurrence Date and Time (Day of Week)</p:item></p:name>
   <p:datatype><p:item>UINT8</p:item></p:datatype>
   <p:range><p:enumRefItem>EnumDayOfWeek</p:enumRefItem></p:range>
   <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(9) Item description of ErrorDetail** element

Table 5-353 Item Description Specifications of ErrorDetail Element**

No.	Item Name	Description Details
1.	LABEL	Write "ErrorDetail**". Write any string in the area marked by the asterisks (**).
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
5.	DATATYPE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of ErrorDeteil1 element

```
<p:commandArgumentMember label="ErrorDetail1">
   <p:label2><p:item>ErrorDetail1</p:item></p:label2>
   <p:name><p:item>Error History Detailed Information 1</p:item></p:name>
   <p:datatype><p:item>WORD</p:item></p:datatype>
   <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

5.5.4 Method Specifications for conducting error checks of equipment

CC-Link and CC-Link IE Field Network define specifications related to METHOD that checks errors of the machine.

5.5.4.1 Overview

Judges whether or not there are errors in the equipment, and if a METHOD is written in order to acquire the error code when an error does occur in the equipment, write this as in Table 5-354. Even on devices for which an error code cannot be acquired, it is possible to write a METHOD that indicates information for the error judgment only. See Table 5-357 for the difference in the description by the acquisition propriety.

LABEL2 and NAME are recommended specifications, and other strings can be written.

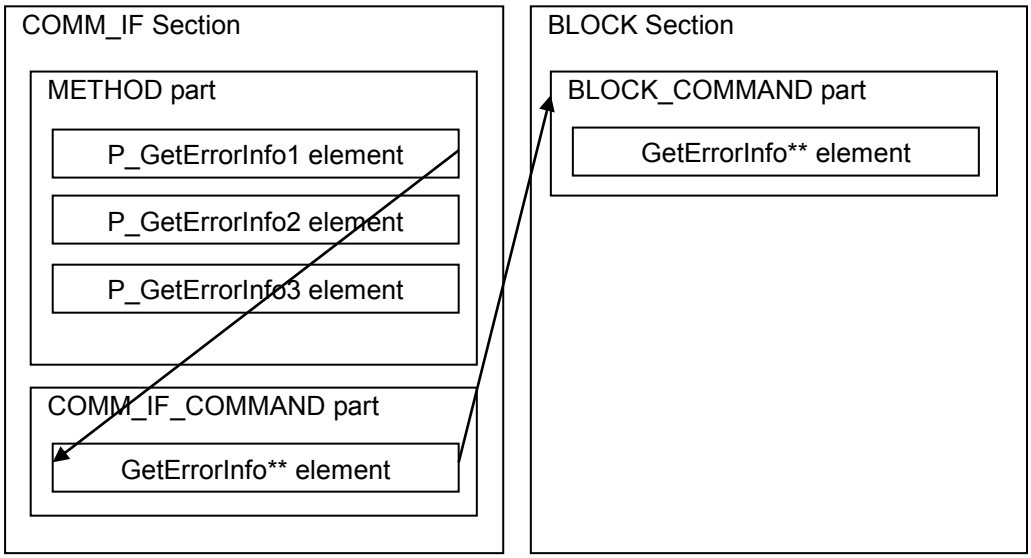


Figure 5-44 Structural image when conducting error check of the equipment

Elements written in METHOD parts are indicated in Table 5-354. See section 5.5.4.2 for details of description specifications of elements.

Table 5-354 List of elements written in METHOD part

No.	Element Name	Description Details	Required/Optional
1.	P_GetErrorInfo*	Writes the method for conducting error check of equipment.	Required

Elements written in COMM_IF_COMMAND parts are indicated in Table 5-355. See section 5.5.4.3 for details of description specifications of elements.

Table 5-355 List of elements written in COMM_IF_COMMAND part

No.	Element Name	Description Details	Required/Optional
1.	(Name Optional)	Writes a command for conducting error check of equipment.	Required

Table 5-356 List of elements written in BLOCK_COMMAND part

No.	Element Name	Description Details	Required/Optional
1.	(Name Optional)	Writes a command for conducting error check of equipment.	Required

5.5.4.2 Description details of elements written in METHOD part

(1) Item description of P_GetErrorInfo+X elements (if error code can be acquired)

Table 5-357 METHOD Specifications for conducting error check of equipment (if error code can be acquired)

No.	Item Name	Description Details
1.	LABEL	"P_GetErrorInfo"+X *X is empty, or write numerical values in order from 1. (max 99) *1 [Ex] P_GetErrorInfo1, P_GetErrorInfo2, …
2.	LABEL2	"P_GetErrorInfo"+X *X is the same as LABEL Using a description that can read 4 digit western calendar years is recommended [Recommended Specifications]
3.	CATEGORY	Not prescribed. Write the same as normal METHOD part.
4.	NAME	Write "Get Error Info" [Recommended Specifications]
5.	TARGET	Writes the label name of COMM_IF_COMMAND prescribed in Table 5-358 for getting error info. *2
6.	METHOD_TYPE	Writes "OTHER" or "COMMAND".
7.	WRITE_REGISTER	Write prohibited.
8.	WRITE_DATA	Write prohibited.
9.	WRITE_DATATYPE	Write prohibited.
10.	READ_REGISTER	Write an address for error code acquisition. Either write the address directly, or reference the ASSIGN element of COMMAND_ARGUMENT referenced through TARGET. *3
11.	READ_DATA	References the VALUE element of COMMAND_ARGUMENT referenced through TARGET. *3
12.	READ_DATATYPE	Writes the data type of the error code. Either write the data type directly or reference the DATATYPE element of COMMAND_ARGUMENT referenced through TARGET.
13.	INTERLOCK	Write the error judgment conditions.
14.	REQ_FLAG	Write prohibited.
15.	END_CONDITION	Write prohibited.
16.	ERR_CONDITION	Write prohibited.
17.	ERR_REGISTER	Write prohibited.
18.	ERR_CODE_RANGE	Write prohibited.
19.	RELATED_METHOD	Write prohibited.
20.	COMMENT	Write prohibited.

*1 If there are multiple error judgment conditions or means of acquiring error codes, by entering different numerical values in X, it is possible to write multiple METHODs. The values of X indicate priority, and if there is an upper limit to the number of error informations that can be displayed with engineering tools, it displays information in favor of that which was acquired with smaller X value.

*2 In the case of equipment for which an error code can be acquired, the error code acquisition argument is defined with the COMMAND_ARGUMENT referenced from the item ARGUMENT of BLOCK_COMMAND referenced by COMM_IF_COMMAND. When writing the error code's meaning (NAME) and comment (COMMENT), ENUM is defined, and they are referenced from the item RANGE of the error code acquisition argument.

*3 Multiple descriptions are not possible. When multiple error codes can be acquired, write them in another P_GetErrorInfoX.

(a) Example item description of P_GetErrorInfo+X element

```
<p:methodMember label=" P_GetErrorInfo1">
   <p:label2><p:item>P_GetErrorInfo1</p:item></p:label2>
   <p:name><p:item>Get Error Info1</p:item></p:name>
   <p:target><p:item> CCLinkCommand.GetErrorInfo</p:item></p:target>
   <p:methodType><p:item>OTHER</p:item></p:methodType>
   <p:interlock><p:item>RX1A==ON</p:item></p:interlock>
   <p:readRegister><p:item>RWr2</p:item></p:readRegister>
   <p:readData><p:item> $(ARGUMENT.ErrorCode.VALUE)</p:item></p:readData>
   <p:readDatatype><p:item> $(ARGUMENT.ErrorCode.DATATYPE)</p:item></p:readDatatype>
</p:methodMember>
```

(2) Example item description of P_GetErrorInfo+X element (when error code cannot be acquired)

Table 5-358 METHOD Specifications for conducting error check of equipment (when error code cannot be acquired)

No.	Item Name	Description Details
1.	LABEL	"P_GetErrorInfo"+X *X is empty, or write numerical values in order from 1. (max 99) *1 [Ex] P_GetErrorInfo1, P_GetErrorInfo2, …
2.	LABEL2	"P_GetErrorInfo"+X *X is the same as LABEL Using a description that can read 4 digit western calendar years is recommended [Recommended Specifications]
3.	CATEGORY	Not prescribed. Write the same as normal METHOD part.
4.	NAME	Write "Get Error Info" [Recommended Specifications]
5.	TARGET	Writes the label name of COMM_IF_COMMAND prescribed in Table 5-359 for getting error info. *2
6.	METHOD_TYPE	Writes "OTHER" or "COMMAND".
7.	WRITE_REGISTER	Write prohibited.
8.	WRITE_DATA	Write prohibited.
9.	WRITE_DATATYPE	Write prohibited.
10.	READ_REGISTER	Write prohibited.
11.	READ_DATA	Write prohibited.
12.	READ_DATATYPE	Write prohibited.
13.	INTERLOCK	Write the error judgment conditions.
14.	REQ_FLAG	Write prohibited.
15.	END_CONDITION	Write prohibited.
16.	ERR_CONDITION	Write prohibited.
17.	ERR_REGISTER	Write prohibited.
18.	ERR_CODE_RANGE	Write prohibited.
19.	RELATED_METHOD	Write prohibited.
20.	COMMENT	Write prohibited.

*1 If there are multiple error judgment conditions or means of acquiring error codes, by entering different numerical values in X, it is possible to write multiple METHODs. The values of X indicate priority, and if there is an upper limit to the number of error informations that can be displayed with engineering tools, it displays information in favor of that which was acquired with smaller X value.

*2 In the case of equipment for which an error code can be acquired, the error code acquisition argument is defined with the COMMAND_ARGUMENT referenced from the item ARGUMENT of BLOCK_COMMAND referenced by COMM_IF_COMMAND. When writing the error code's meaning (NAME) and comment (COMMENT), ENUM is defined, and they are referenced from the item RANGE of the error code acquisition argument.

5.5.4.3 Example of description of elements written in COMM_IF_COMMAND part

Not prescribed.

Write the same as normal COMM_IF_COMMAND. An example description is shown below.

Table 5-359 Example item description of elements of COMM_IF_COMMAND part

No.	Item Name	Description Details
1.	LABEL	"GetErrorCode" is written.
2.	LABEL2	Not written because it may be omitted.
3.	CATEGORY	Not written because it may be omitted.
4.	NAME	"Get Error Info Command" is written.
5.	ARGUMENT	Not written because it may be omitted.
6.	REF	"BlockSec.BlockCommand.GetErrorInfo" is written.
7.	COMMENT	Not written because it may be omitted.

(a) Example of item description of GetErrorInfo element

```
<p:commIfCommandMember label="GetErrorInfo">
   <p:name><p:item>Get Error Info Command</p:item></p:name>
   <p:ref><p:item>BlockSec.BlockCommand.GetErrorInfo</p:item></p:ref >
</p:commIfCommandMember >
```

5.5.4.4 Description details of element written in BLOCK_COMMAND part

(1) Item description of BLOCK_COMMAND element

Table 5-360 Item Description Specifications of BLOCK_COMMAND Element

No.	Item Name	Description Details
1.	LABEL	"GetErrorInfo1" is written.
2.	LABEL2	Not written because it may be omitted.
3.	CATEGORY	Not written because it may be omitted.
4.	NAME	Not prescribed. Write the same as normal BLOCK_COMMAND parts.
5.	ARGUMENT	Writes a reference to the COMMAND_ARGUMENT in which the elements prescribed in Table 5-356 are written.
6.	COMMENT	Not written because it may be omitted.

(a) Item description example of GetErrorLogCommand element

```
<p:blockCommandMember label="GetErrorLogCommand">
   <p:name><p:item>Get Error Info Command</p:item></p:name>
   <p:argument><p:item>ErrorInfoArgument</p:item></p:argument>
</p:blockCommandMember>
```

5.5.4.5 Description details of elements written in COMMAND_ARGUMENT part

(1) Item description of ErrorCode elements

Table 5-361 Item description example of ErrorCode element

No.	Item Name	Description Details
1.	LABEL	"ErrorCode" is written.
2.	LABEL2	Not written because it may be omitted.
3.	CATEGORY	Not written because it may be omitted.
4.	NAME	Not prescribed. Write the same as normal COMMAND_ARGUMENT parts.
5.	DATATYPE	WORD is written.
6.	DEFAULT	Not written because it may be omitted.
7.	RANGE	"ENUM EnumErrorCode" is written.
8.	MIN_INC	Not written because it may be omitted.
9.	ENG_UNIT	Not written because it may be omitted.
10.	ACCESS	"R" is written.
11.	ASSIGN	Not written because it may be omitted.
12.	REF	Not written because it may be omitted.
13.	COMMENT	Not written because it may be omitted.

(a) Item description example of ErrorCode element

```
<p:commandArgumentMember label="ErrorCode">
   <p:name><p:item>Error Code</p:item></p:name>
   <p:datatype><p:item>WORD</p:item></p:datatype>
   <p:range><p:enumRefItem>ENUM EnumErrorCode</p:enumRefItem></p:range>
   <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

METHOD

P_GetErrorInfo1

	LABEL	NAME	TARGET	METHOD_TYPE	READ_REGISTER	READ_DATA	READ_DATATYPE	INTERLOCK
1	P_GetErrorInfo1	エラー情報取得1	CCLinkCommand.GetErrorInfo	OTHER	RWr2	$(ARGUMENT.ErrorCode.VALUE)	$(ARGUMENT.ErrorCode.DATATYPE)	RX1A==ON

COMM_IF_COMMAND (LABEL=CCLinkCommand)

CCLinkCommand

	LABEL	NAME	ARGUMENT	REF
1	GetErrorInfo	エラー情報取得コマンド	ErrorInfoArgument	BlockSec.BlockCommand.GetErrorInfo

BLOCK_COMMAND (LABEL=BlockCommand)

BlockCommand

	LABEL	NAME	ARGUMENT
1	GetErrorInfo	エラー取得コマンド	ErrorInfoArgument

COMMAND_ARGUMENT (LABEL=ErrorInfoArgument)

ErrorInfoArgument

	LABEL	NAME	DATATYPE	RANGE	ACCESS
1	ErrorCode	エラーコード	WORD	ENUM EnumErrorCode	R

ENUM (LABEL=EnumErrorCode)

EnumErrorCode

	LABEL	NAME	CODE	COMMENT1
1	enum0	ハードウェアエラー	0x0	電源のOFF→ONを再び行ってください。再度発生する場合は、ユニットの故障が考えられます。本フラグがONした場合...
2	enum1	ユーザレンジ読み出しエラー	0x1	ユーザレンジにエラーがあった場合にONします。エラーリセット要求フラグをONすることで本フラグをリセット(OFF)できます。
3	enum2	フラッシュメモリ書き込みエラー	0x2	オフセット・ゲイン設定時に、フラッシュメモリへの書き込みが失敗、もしくはフラッシュメモリへの書込みに失敗した場合にO...

Figure 5-45 Examples of METHOD & COMMAND that perform error check of equipment (some unwritable or optional items omitted)

5.5.5 Parts and elements that indicated parameters for safety communication function

The safety compatible devices of CC-Link IE Field Network define specifications related to parameters for the safety communication function.

This section describes specifications of parts and elements for writing the parameters of safety communication functions.

5.5.5.1 Overview

Write safety communication function parameters in COMM_IF_PARAMETER parts and BLOCK_PARAMETER parts.

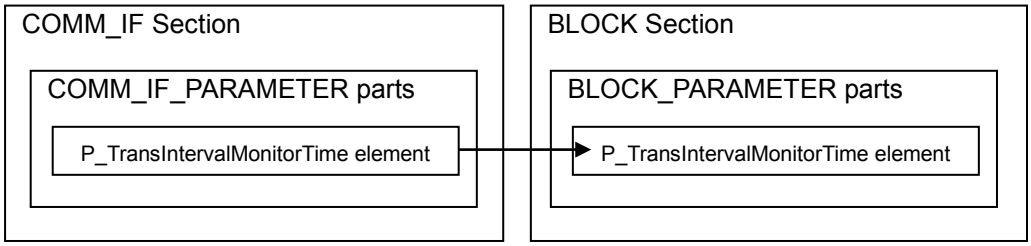


Figure 5-46 Structural image in when writing parameters of safety communication functions

Elements written in COMM_IF_PARAMETER parts are indicated in Table 5-. See Section 5.5.5.2 for details of description specifications of elements.

Table 5-362 Example of item description in the case of ErrorCode elements

No.	Element Name	Description Details	Required/Optional
1.	P_TransIntervalMonitorTime	Describes the transmission interval monitoring time of safety communication function.	Required

Elements written in BLOCK_PARAMETER parts are indicated inTable 5-363. See Section 5.5.5.2 for details of description specifications of elements.

Table 5-363 List of elements written in BLOCK_PARAMETER parts

No.	Element Name	Description Details	Required/Optional
1.	P_TransIntervalMonitorTime	Describes the transmission interval monitoring time of safety communication function.	Required

5.5.5.2 Description details of elements written in COMM_IF_PARAMETER parts and BLOCK_PARAMETER parts

(1) Item description of elements of P_TransIntervalMonitorTime of COMM_IF_PARAMETER parts

Table 5-364 Item Description Specifications of P_TransIntervalMonitorTime Element of COMM_IF_PARAMETER Part

No.	Item Name	Description Details
1.	LABEL	Write "P_TransIntervalMonitorTime".
2.	LABEL2	Write "P_TransIntervalMonitorTime".
3.	CATEGORY	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.
4.	NAME	Write "Transmission Interval Monitoring Time".
5.	DATATYPE	May be omitted because this case uses the value of the reference source of REF item. When writing, "UINT16" is written.
6.	DEFAULT	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.
7.	RANGE	May be omitted because this case uses the value of the reference source of REF item. When writing, " [2,65535] " is written.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	May be omitted because this case uses the value of the reference source of REF item. When writing, "ms" is written.
10.	ACCESS	May be omitted because this case uses the value of the reference source of REF item. When writing, "W" is written.
11.	WRITE_ORDER	Write prohibited.
12.	ASSIGN	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.
13.	UI_ATTRIBUTE	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.
14.	REF	Write a reference to the "P_TransIntervalMonitorTime" of BLOCK_PARAMETER parts.
15.	COMMENT	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.

(a) Example of item description of elements

```
<p:commIfParameterMember label="P_TransIntervalMonitorTime">
   <p:label2><p:item>P_TransIntervalMonitorTime</p:item></p:label2>
   <p:name><p:item>Transmission Interval Monitoring Time</p:item></p:name>
   <p:ref><p:item>BlockSec.BlockParameter.P_TransIntervalMonitorTime</p:item></p:ref>
</p:commIfParameterMember>
```

(2) Item description of P_TransIntervalMonitorTime elements of BLOCK_PARAMETER parts

Table 5-365 Item Description Specifications of P_TransIntervalMonitorTime Element of BLOCK_PARAMETER Part

No.	Item Name	Description Details
1.	LABEL	Write "P_TransIntervalMonitorTime".
2.	LABEL2	Write "P_TransIntervalMonitorTime".
3.	CATEGORY	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.
4.	NAME	Write "Transmission Interval Monitoring Time".
5.	DATATYPE	Write "UINT16".
6.	DEFAULT	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.
7.	RANGE	Not prescribed. Write the same as normal BLOCK_PARAMETER parts. However, because communication interval monitoring time is 16 bit in 128μs units in protocol specifications, values above 128×65535÷1000≒8388ms cannot be specified.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write "ms".
10.	ACCESS	Write "W" or "RW".
11.	WRITE_ORDER	Write prohibited.
12.	UI_ATTRIBUTE	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.
13.	COMMENT	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.

(a) Examples of item description of elements

```
<p:blockParameterMember label="P_TransIntervalMonitorTime">
   <p:label2><p:item>P_TransIntervalMonitorTime</p:item></p:label2>
   <p:name><p:item>Transmission Interval Monitoring Time</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:range><p:item>[10,2000]</p:item></p:range>
   <p:engUnit><p:item>ms</p:item></p:engUnit>
   <p:access><p:item>RW</p:item></p:access>
</p:blockParameterMember>
```

5.5.6 MESSAGE Specifications for parameter writing of safety communication functions

The safety compatible devices of CC-Link IE Field Network define specifications related to parameter writing of the safety communication function.

5.5.6.1 Overview

When writing a MESSAGE to write parameters in safety equipment, write it according to the specifications prescribed in this section. However, because LABEL2 and NAME are recommended specifications, other strings may be written.

Also, when reading CRC32 values, it writes the strings indicated in Table 5-366 in REQUEST_DATA and RESPONSE_DATA. Parameters of safety communication functions (See Section 5.5.5) are also written using messages shown in this section.

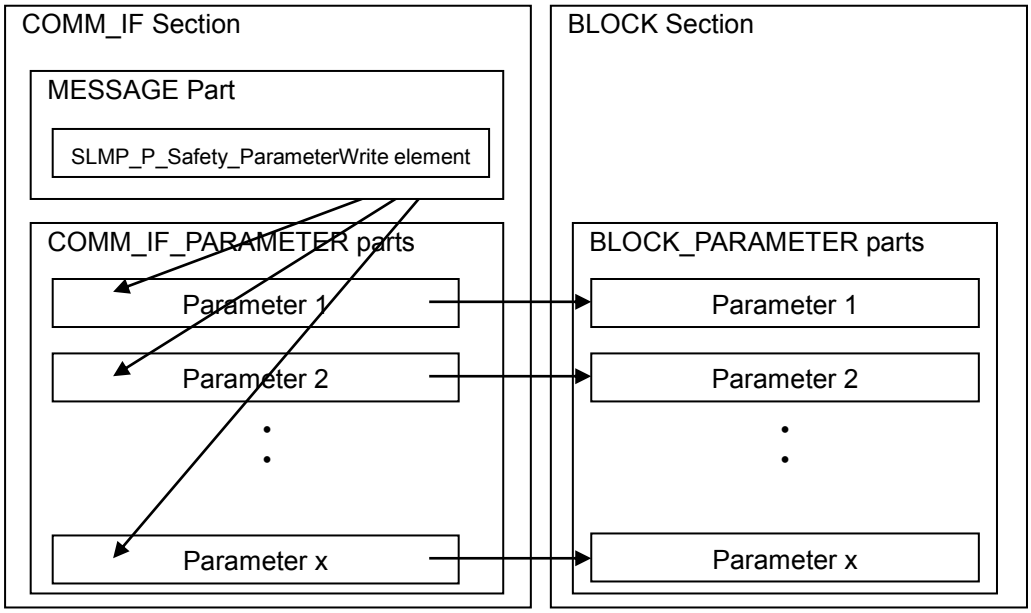


Figure 5-47 Structural image when describing parameter writing of safety communication functions

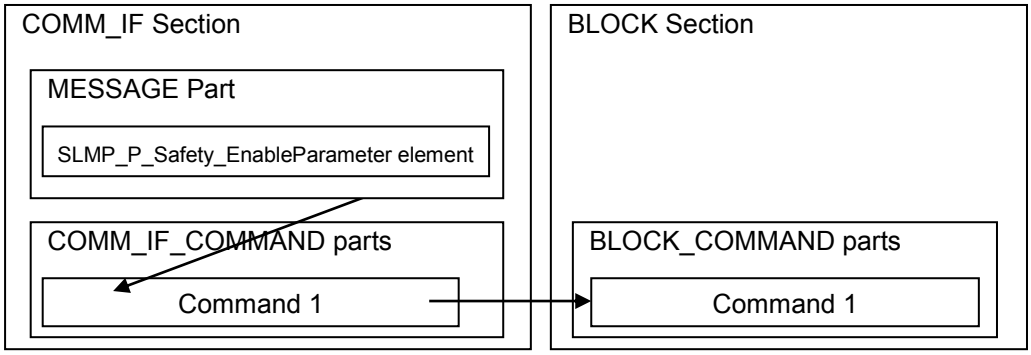


Figure 5-48 Structural image when describing enabling parameters of safety communication functions

Table 5-366 Reserved words that indicated read/write of CRC32 values

Name	Reserved
CRC32 value prescribed in CC-Link IE Safety Communication Function *1	$(CRC32_F1922815)
CRC32 value prescribed in IEEE802.3. *1	$(CRC32_04C11DB7)

*1 Utility software calculates CRC32 according to the following specifications.

・When running a MESSAGE element including reserved words indicating CRC32, CRC32 is calculated from the VALUE of all elements of COMM_IF_PARAMETER (BLOCK_PARAMETER) referenced by the TARGET of that MESSAGE element.

・When running multiple MESSAGE elements sequentially with RELATED_MESSAGE (in the case that it includes reserved words indicating CRC32 in at least one of those elements), CRC32 is calculated from the VALUE of all elements of COMM_IF_PARAMETER (BLOCK_PARAMETER) referenced by the TARGET of all MESSAGE elements referenced by RELATED_MESSAGE.

When reserved words indicating the value of CRC32 are written in REQUEST_DATA, utility software calculates CRC32 when MESSAGE is executed, and stores it in a request message. If it is written in RESPONSE_DATA, the utility software reads the value of CRC32 from the place where the reply message reserved word is written.

When parameter writing of safety communication functions is described with multiple consolidated MESSAGES, elements prescribed in this section (SLMP_P_Safety_ParameterWrite) it is attached to a MESSAGE representing a series of operations. The elements of the MESSAGE consolidated from there are optional.

A maximum of 1 reserved word of Table 5-366 can be written in each REQUEST_DATA or RESPONSE_DATA.

Elements written in MESSAGE parts are indicated in Table 5-367. See Section 5.5.6.2 for details of description specifications of elements.

Table 5-367 List of elements written in MESSAGE parts

No.	Element Name	Description Details	Required/Optional
1.	SLMP_P_Safety_ParameterWrite	Writes the message used in parameter writing of safety communication functions.	Optional
2.	SLMP_P_Safety_EnableParameter	Writes the enabling information used in parameter writing of safety communication functions.	Optional

(1) Elements of parameters referenced by SLMP_P_Safety_ParameterWrite elements

Elements written in COMM_IF_PARAMETER parts are indicated in Table 5-368. See Section 5.5.6.2 for details of description specifications of elements.

Table 5-368 List of elements written in COM_IF_PARAMETER parts

No.	Element Name	Description Details	Required/Optional
1.	(Name Optional)	Writes the parameters of safety communication functions.	Optional

Elements written in BLOCK_PARAMETER parts are indicated in Table 5-369. See Section 5.5.6.2 for details of description specifications of elements.

Table 5-369 List of elements written in BLOCK_PARAMETER parts

No.	Element Name	Description Details	Required/Optional
1.	(Name Optional)	Writes the parameters of safety communication functions.	Optional

(2) Elements of commands referenced by SLMP_P_Safety_EnableParameter elements

Elements written in COMM_IF_COMMAND parts are indicated in Table 5-370. See Section 5.5.6.2 for details of description specifications of elements.

Table 5-370 List of elements written in BLOCK_PARAMETER parts

No.	Element Name	Description Details	Required/Optional
1.	(Name Optional)	Writes the parameters of safety communication functions.	Optional

Elements written in BLOCK_COMMAND parts are indicated in Table 5-371. See Section 5.5.6.2 for details of description specifications of elements.

Table 5-371 List of elements written in BLOCK_PARAMETER parts

No.	Element Name	Description Details	Required/Optional
1.	(Name Optional)	Writes the parameters of safety communication functions.	Optional

5.5.6.2 MESSAGE specifications for parameter writing of safety communication functions

MESSAGEs are prescribed for parameter writing, but not for reading. MESSAGEs for reading can be written freely according to CSP+ specifications.

(1) Item description of SLMP_P_Safety_ParameterWrite elements

Table 5-372 Item Description Specifications of SLMP_P_Safety_ParameterWrite Element

No.	Item Name	Description Details
1.	LABEL	Write "SLMP_P_Safety_ParameterWrite".
2.	LABEL2	Write "SLMP_P_Safety_ParameterWrite". [Recommended Specification]
3.	CATEGORY	Not prescribed. Write the same as normal MESSAGE parts.
4.	NAME	Write "Parameter Writing". [Recommended Specification]
5.	TARGET	Write the reference to the COMM_IF_PARAMETER written in safety communication function parameters.
6.	MESSAGE_TYPE	Write "PARAMETER".
7.	REQUEST_TYPE	Not prescribed. Write the same as normal MESSAGE parts.
8.	REQUEST_DATA	Not prescribed. Write the same as normal MESSAGE parts.
9.	REQUEST_DATATYPE	Not prescribed. Write the same as normal MESSAGE parts.
10.	RESPONSE_TYPE	Not prescribed. Write the same as normal MESSAGE parts.
11.	RESPONSE_DATA	Not prescribed. Write the same as normal MESSAGE parts.
12.	RESPONSE_DATATYPE	Not prescribed. Write the same as normal MESSAGE parts.
13.	ERR_TYPE	Not prescribed. Write the same as normal MESSAGE parts.
14.	ERR_CODERANGE	Not prescribed. Write the same as normal MESSAGE parts.
15.	RELATED_MESSAGE	Not prescribed. Write the same as normal MESSAGE parts.
16.	COMMENT	Not prescribed. Write the same as normal MESSAGE parts.

(a) Item description example of SLMP_P_Safety_ParameterWrite element

```
<p:messageMember label="SLMP_P_Safety_ParameterWrite">
   <p:label2><p:item>SLMP_P_Safety_ParameterWrite</p:item></p:label2>
   <p:name><p:item>Parameter Writing</p:item></p:name>
   <p:target><p:item>SEQ TARGET</p:item></p:target>
   <p:messageType><p:item>PARAMETER</p:item></p:messageType>
   <p:relatedMessage>
      <p:item>&lt;SEQ SLMPStartWritePramArea&gt;&lt;SEQ SLMPStationWritePrm&gt;&lt;SEQ
      SLMPWriteCRC32&gt;&lt;SEQ SLMPReflectPrm&gt;&lt;SEQ SLMPStartReadPramArea&gt;&lt;SEQ
      SLMPReadCRC32&gt;</p:item>
   </p:relatedMessage>
</p:messageMember>
```

(2) Item description of SLMP_P_Safety_EnableParameter elements

Table 5-373 Item Description Specifications of SLMP_P_Safety_EnableParameter Element

No.	Item Name	Description Details
1.	LABEL	Write "SLMP_P_Safety_EnableParameter".
2.	LABEL2	Write "SLMP_P_Safety_EnableParameter". [Recommended Specification]
3.	CATEGORY	Not prescribed. Write the same as normal MESSAGE parts.
4.	NAME	Write "Enabling Parameter". [Recommended Specification]
5.	TARGET	Write the reference to the COMM_IF_COMMAND written in safety communication function parameters.
6.	MESSAGE_TYPE	Write "COMMAND".
7.	REQUEST_TYPE	Not prescribed. Write the same as normal MESSAGE parts.
8.	REQUEST_DATA	Not prescribed. Write the same as normal MESSAGE parts.
9.	REQUEST_DATATYPE	Not prescribed. Write the same as normal MESSAGE parts.
10.	RESPONSE_TYPE	Not prescribed. Write the same as normal MESSAGE parts.
11.	RESPONSE_DATA	Not prescribed. Write the same as normal MESSAGE parts.
12.	RESPONSE_DATATYPE	Not prescribed. Write the same as normal MESSAGE parts.
13.	ERR_TYPE	Not prescribed. Write the same as normal MESSAGE parts.
14.	ERR_CODERANGE	Not prescribed. Write the same as normal MESSAGE parts.
15.	RELATED_MESSAGE	Not prescribed. Write the same as normal MESSAGE parts.
16.	COMMENT	Not prescribed. Write the same as normal MESSAGE parts.

(a) Item description example of SLMP_P_Safety_EnableParameter element

```
<p:messageMember label="SLMP_P_Safety_EnableParameter">
   <p:label2><p:item>SLMP_P_Safety_EnableParameter</p:item></p:label2>
   <p:name><p:item>Enabling Parameter</p:item></p:name>
   <p:target><p:item>CommCommand.Safety_EnableParameter</p:item></p:target>
   <p:messageType><p:item>COMMAND</p:item></p:messageType>
   <p:requestType><p:item>wrReqMT_Binary</p:item></p:requestType>
   <p:requestData>
      <p:item>&lt;0x1613&gt;&lt;0x0000&gt;&lt;0x0000FFFF&gt;&lt;0x0001&gt;&lt;0xFFFE&gt
      ;</p:item>
   </p:requestData>
   <p:requestDatatype>
      <p:item>&lt;WORD&gt;&lt;WORD&gt;&lt;DWORD&gt;&lt;WORD&gt; &lt;WORD&gt;</p:item>
   </p:requestDatatype>
   <p:responseType><p:item>wrResMT_Binary</p:item></p:responseType>
</p:messageMember>
```

Figure 5-49 shows examples of MESSAGE for parameter writing of safety communication functions.

LABEL	LABEL2	CATEGORY	NAME	TARGET	ERR_CODE_RANGE
SLMP P Safety ParameterWrite	パラメータ書込み		パラメータ書込み	SEQ TARGET	
SLMPStationWritePrm	パラメータ書込み 局パラメータ		パラメータ書込み(局パラメータ)	StationParam.*	
SLMPBasicUnitWritePrm	パラメータ書込み 基本ユニット		パラメータ書込み(基本ユニット)	BasicUnitParam.*	
SLMPReflectPrm	パラメータ設定要求		パラメータ設定要求	CommCommand.ReflectPrmCommand	
SLMP P Safety EnableParameter	パラメータ有効化		パラメータ有効化	SEQ TARGET	
SLMPWriteCRC32	CRC32書込み		CRC32書込み	CommCommand.WriteCRC32Command	
SLMPCheckCRC32	CRC32照合		CRC32照合	CommCommand.CheckCRC32Command	
SLMPReadCRC32	CRC32読出し		CRC32読出し	CommCommand.ReadCRC32Command	
SLMPSafetyEnableParameter	安全ユニット有効化		安全ユニット有効化	CommCommand.Safety EnableParameterCommand	

MESSAGE_TYPE	REQUEST_TYPE	REQUEST_DATA	REQUEST_DATATYPE	RESPONSE_TYPE
PARAMETER				
OTHER	wrReqMT Binary	<0x1613><0x0000><0x00000000><0x0006><$(*.VALUE)>	<WORD><WORD><DWORD><WORD><$(*.DATATYPE)>	wrResMT Binary
OTHER	wrReqMT Binary	<0x1613><0x0000><0x00000100><0x00D3><$(*.VALUE)>	<WORD><WORD><DWORD><WORD><$(*.DATATYPE)>	wrResMT Binary
OTHER	wrReqMT Binary	<0x1613><0x0000><0x0000FFFF><0x0001><0xFFFC>	<WORD><WORD><DWORD><WORD><WORD>	wrResMT Binary
COMMAND				
OTHER	wrReqMT Binary	<0x1613><0x0000><0x0000E000><0x0002><$(CRC32 F1922815)>	<WORD><WORD><DWORD><WORD><DWORD>	wrResMT Binary
OTHER	wrReqMT Binary	<0x1613><0x0000><0x0000FFFF><0x0001><0xFFFD>	<WORD><WORD><DWORD><WORD><WORD>	wrResMT Binary
OTHER	rdReqMT Binary	<0x0613><0x0000><0x0000E002><0x0002>	<WORD><WORD><DWORD><WORD>	rdResMT Binary
COMMAND	wrReqMT Binary	<0x1613><0x0000><0x0000FFFF><0x0001><0xFFFB>	<WORD><WORD><DWORD><WORD><WORD>	wrResMT Binary

RESPONSE_DATA	RESPONSE_DATATYPE	ERR_TYPE	RELATED_MESSAGE	COMMENT
			<SEQ SLMPStationWritePrm><SEQ SLMPBasicUnitWritePrm><SEQ SLMPReflectPrm>	
			<SEQ SLMPWriteCRC32><SEQ SLMPCheckCRC32><SEQ SLMPReadCRC32><SEQ SLMPSafetyEnableParameter>	
		wrErrMT Binary		
		wrErrMT Binary		
<$(CRC32 04C11DB7)>	<DWORD>	rdErrMT Binary		
		wrErrMT Binary		

Figure 5-49 Example of MESSAGE for parameter writing of safety communication functions

5.5.7 Conventions of read/write capable parameters using parameter IDs

SLMP defines specifications related to parameters that can be written using parameter IDs.

5.5.7.1 Overview

This section shows conventions of parameters which can read and write using SLMP and parameter IDs. This convention, in the current version, can only be written in CSP+ for the Ethernet connection unit. A structural image of the parameters using this convention are shown in Figure 5-50.

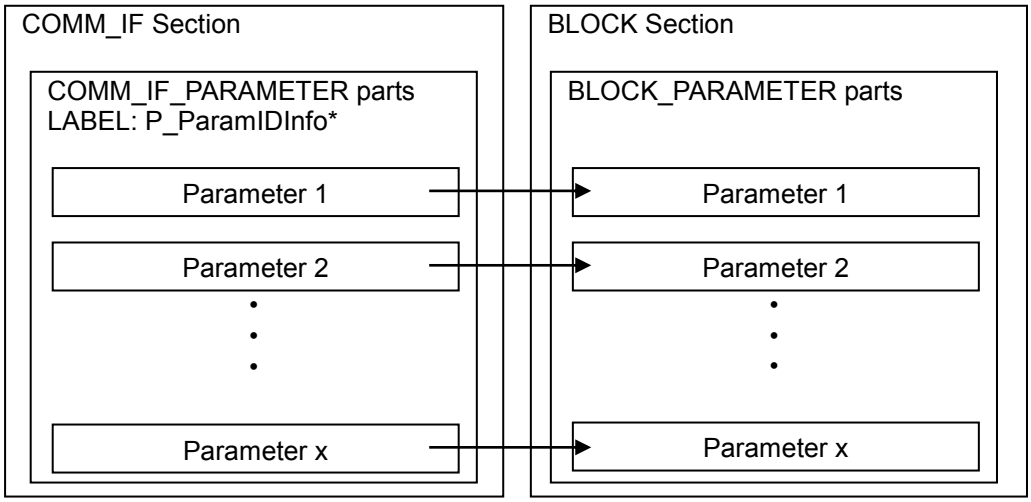


Figure 5-50 Structural image of read/write capable parameters using parameter IDs

Parameters written with this convention are written in COMM_IF_PARAMETER parts with the prefix "P_ParamIDInfo" in its LABEL. In this convention, description by MESSAGE parts is not used. Reading of parameters written with this convention are conducted using the GetParameter command of SLMP, while writing is done in a series of steps using SetParameter related commands of SLMP.

(1) Conventions of COMM_IF_PARAMETER parts

Conventions of COMM_IF_PARAMETER parts are shown in Table 5-374.

Table 5-374 Conventions of COMM_IF_PARAMETER parts

No.	Part Name	Description Details	Required/Optional
1.	P_ParamIDInfo*	Writes parameters that perform read/write using parameter ID.	Optional

With COM_IF_PARAMETER parts written with this convention, as the value of labels which are an attribute XML elements indicating parts, any strings starting with "P_ParamIDInfo" are written.

(2) Elements of parameters written in P_ParamIDInfo* parts

Elements written in COMM_IF_PARAMETER parts are indicated in Table 5-375. See Section 5.5.7.2 for details of description specifications of elements.

Table 5-375 List of elements written in P_ParamIDInfo*

No.	Element Name	Description Details	Required/Optional
1.	(Name Optional)	Writes parameters that perform read/write using parameter ID.	Optional

Elements written in BLOCK_PARAMETER parts are indicated in Table 5-376. See Section 5.5.7.2 for details of description specifications of elements.

Table 5-376 List of elements written in BLOCK_PARAMETER parts

No.	Element Name	Description Details	Required/Optional
1.	(Name Optional)	Writes parameters that perform read/write using parameter ID.	Optional

5.5.7.2 Description specifications of parameters which perform read/write using parameter ID

Conventions of parameters which perform read/write using parameter ID are shown here.

(1) Item descriptions of elements of COMM_IF_PARAMETER

Item description specifications of elements written in COMM_IF_PARAMETER parts are indicated in Table 5-377.

Table 5-377 Item description specifications of COMM_IF_PARAMETER elements

No.	Item Name	Description Details
1.	LABEL	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.
2.	LABEL2	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.
3.	CATEGORY	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.
4.	NAME	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.
5.	DATATYPE	When word data (byte data) which one parameter ID is assigned to is divided and assigned a different meaning for each bit, it writes a reference to STRUCT parts. Otherwise it is recommended that this be omitted.
6.	DEFAULT	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.
7.	RANGE	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.
8.	MIN_INC	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.
9.	ENG_UNIT	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.
10.	ACCESS	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.
11.	WRITE_ORDER	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.
12.	ASSIGN	Writes parameter ID in hexadecimal numbers. Does not write "0x" to indicated hexadecimal numbers.
13.	UI_ATTRIBUTE	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.
14.	REF	Writes a reference to elements of BLOCK_PARAMETER parts.
15.	COMMENT	Not prescribed. Write the same as normal COMM_IF_PARAMETER parts.

(a) Item description example of COMM_IF_PARAMETER part element

```
<p:commIfParameter label="P_ParamIDInfo 1">
   <p:commIfParameterMember label="Param1">
      <p:label2><p:item>Parameter 1</p:item></p:label2>
      <p:name><p:item>Parameter 1</p:item></p:name>
      <p:assign><p:item>0070</p:item></p:assign>
      <p:ref><p:item>BlockSec.BlockParameter.Param1</p:item></p:ref>
   </p:commIfParameterMember>
</p:commIfParameter>
```

(2) Item description of elements of BLOCK_PARAMETER parts

Item description specifications of elements written in COMM_IF_PARAMETER parts are shown in Table 5-378.

Table 5-378 Item description specifications of elements of BLOCK_PARAMETER parts

No.	Item Name	Description Details
1.	LABEL	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.
2.	LABEL2	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.
3.	CATEGORY	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.
4.	NAME	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.
5.	DATATYPE	When word data (byte data) which one parameter ID is assigned to is divided and assigned a different meaning for each bit, it writes a reference to STRUCT parts. Otherwise it is recommended that this be omitted.
6.	DEFAULT	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.
7.	RANGE	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.
8.	MIN_INC	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.
9.	ENG_UNIT	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.
10.	ACCESS	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.
11.	WRITE_ORDER	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.
12.	UI_ATTRIBUTE	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.
13.	COMMENT	Not prescribed. Write the same as normal BLOCK_PARAMETER parts.

(a) Item description example of BLOCK_PARAMETER part element

```
<p:blockParameterMember label="Param1">
   <p:label2><p:item>Parameter 1</p:item></p:label2>
   <p:name><p:item>Parameter 1</p:item></p:name>
   <p:default><p:item>0</p:item></p:default>
   <p:range><p:item>[0,65535]</p:item></p:range>
   <p:access><p:item>RW</p:item></p:access>
</p:blockParameterMember>
```

5.5.8 Parts and elements related to the acquisition of energy management information

5.5.8.1 Overview

The acquisition of energy management information is described in the METHOD part or MESSAGE part, COMM_IF_COMMAND part, BLOCK_COMMAND part, and COMMAND_ARGUMENT part. Energy management information is sometimes managed on the module main body side and sometimes on the communication module side, depending on the module mode and specifications. The parts that are described, therefore, differ accordingly (parts can be described based on the concept of the COMM_IF section and BLOCK section). Figure 5-51 and Figure 5-52 show images of the structure when the acquisition of energy management information is described.

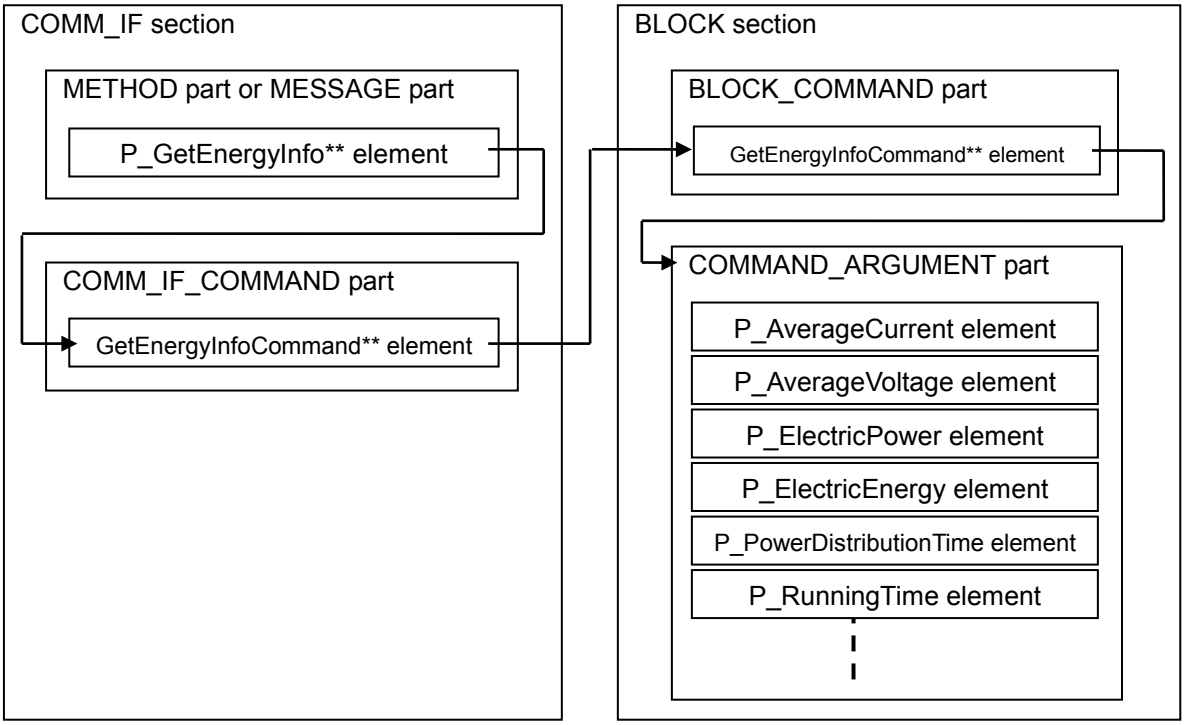


Figure 5-51 Image of Structure When Acquisition of Energy Management Information Is Described (When Energy Management Information Is Managed on Module Main Body Side)

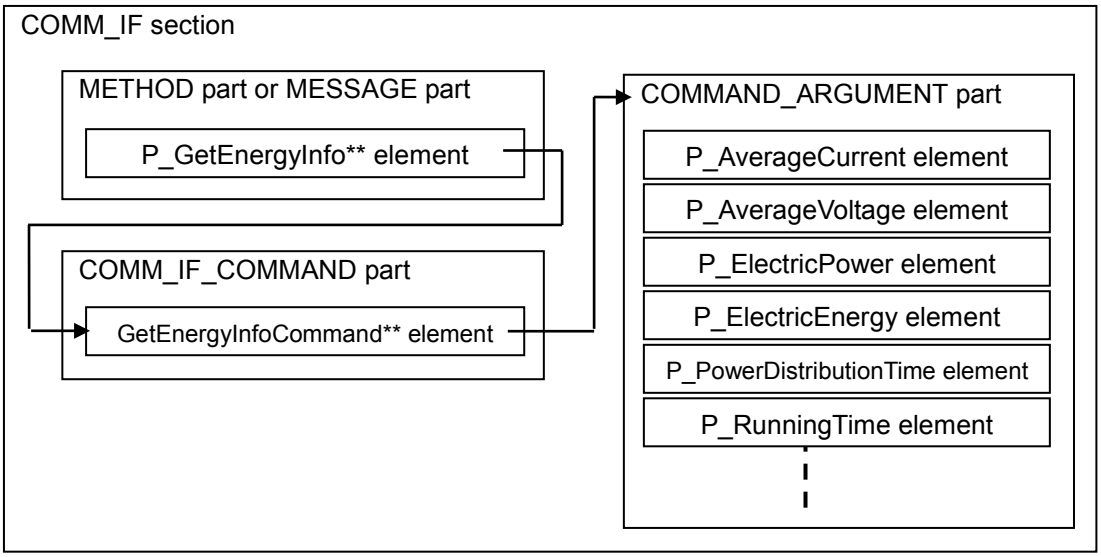


Figure 5-52 Image of Structure When Acquisition of Energy Management Information Is Described (When Energy Management Information Is Managed on Communication Module Side)

Table 5-379 shows a list of elements described in the METHOD part. For details of the description specifications of the elements, see Section 5.5.8.3. Any element is not described when the MESSAGE part is used. Either the METHOD part or MESSAGE part is used.

Table 5-379 List of Elements Described in METHOD Part

No.	Element Name	Description Details	Required/Optional
1.	P_GetEnergyInfo**	Describes the method used to acquire the energy management information. For "**", see detailed description specifications.	Required*1

*1: Required only when the METHOD part is used. Optional when the METHOD part is not used.

Table 5-380 lists the elements described in the MESSAGE part. For details of the description specifications of the elements, see Section 5.5.8.3. Any element is not described when the METHOD part is used. Either the METHOD part or MESSAGE part is used.

Table 5-380 List of Elements Described in MESSAGE Part

No.	Element Name	Description Details	Required/Optional
1.	P_GetEnergyInfo**	Describes the message used to acquire the energy management information. For "**", see detailed description specifications.	Required*2

*2: Required only when the MESSAGE part is used. Optional when the MESSAGE part is not used.

Table 5-381 lists the elements described in the COMM_IF_COMMAND part. For details of the description specifications of the elements, see Section 5.5.8.4.

Table 5-381 List of Elements Described in COMM_IF_COMMAND Part

No.	Element Name	Description Details	Required/Optional
1.	GetEnergyInfoCommand**	Describes the command information used to acquire the energy management information. For "**", see detailed description specifications.	Required

Table 5-382 lists the elements described in the BLOCK_COMMAND part. For details of the description specifications of the elements, see Section 5.5.8.5.

Table 5-382 List of Elements Described in BLOCK_COMMAND Part

No.	Element Name	Description Details	Required/Optional
1.	GetEnergyInfoCommand**	Describes the command information used to acquire the energy management information. For "**", see detailed description specifications.	Required*3

*3: Required only when the BLOCK_COMMAND part is used. Optional when the BLOCK_COMMAND part is not used.

Table 5-383 lists the elements described in the COMMAND_ARGUMENT part (specifications common to COMM_IF and BLOCK sections). We recommend writing "GetEnergyInfoArgument" for the label name of the COMMAND_ARGUMENT part. While all elements are optionally described, at least one element must be described (A description of only the COMMAND_ARGUMENT part in which these elements are not described is not permitted). For details of the description specifications of the elements, see Section 5.5.8.6.

Table 5-383 List of Elements Described in COMMAND_ARGUMENT Part

No.	Element Name	Description Details	Required/Optional
1.	P_AverageCurrent	Total current	Optional
2.	P_AverageVoltage	Total line voltage	Optional
3.	P_ElectricPower	Electric power	Optional
4.	P_ElectricEnergy	Electric energy	Optional
5.	P_PowerDistributionTime	Integrated energization time	Optional
6.	P_RunningTime	Actual operating time	Optional
7.	P_1_PhaseCurrent	One-phase current	Optional
8.	P_2_PhaseCurrent	Two-phase current	Optional
9.	P_3_PhaseCurrent	Three-phase current	Optional
10.	P_1_2lineVoltage	Line voltage between the first phase and the second phase	Optional
11.	P_2_3lineVoltage	Line voltage between the second phase and the third phase	Optional
12.	P_3_1lineVoltage	Line voltage between the third phase and the first phase	Optional
13.	P_1_PhaseCurrentDemand	One-phase demand current	Optional
14.	P_2_PhaseCurrentDemand	Two-phase demand current	Optional
15.	P_3_PhaseCurrentDemand	Three-phase demand current	Optional
16.	P_ElectricPowerDemand	Demand electric power	Optional
17.	P_PowerFactor	Power factor	Optional
18.	P_Frequency	Frequency	Optional
19.	P_ReactiveEnergy	Reactive power	Optional
20.	P_1_0PhaseVoltage	One-phase voltage	Optional
21.	P_2_0PhaseVoltage	Two-phase voltage	Optional
22.	P_3_0PhaseVoltage	Three-phase voltage	Optional

Note that while multiple parts are used to describe the acquisition of energy management information, there are no regulations regarding the label name of any part other than the COMMAND_ARGUMENT part. (The CSP+ file creator can freely assign label names.)

5.5.8.2 Description details of elements described in METHOD part

(1) Item description of P_GetEnergyInfo** element

Table 5-384 Item Description Specifications of P_GetEnergyInfo Element**

No.	Item Name	Description Details
1.	LABEL	Write "P_GetEnergyInfo**". Write any string, including spaces, in the area marked by the asterisks (**). (Example: P_GetEnergyInfoMethod)
2.	LABEL2	Not specified. Write a value in the same manner as the regular METHOD part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular METHOD part.
4.	NAME	Not specified. Write a value in the same manner as the regular METHOD part.
5.	TARGET	Write a reference to the elements of the COMM_IF_COMMAND part specified by Section 5.5.8.4.
6.	METHOD_TYPE	Write "COMMAND".
7.	WRITE_REGISTER	Not specified. Write a value in the same manner as the regular METHOD part.
8.	WRITE_DATA	Not specified. Write a value in the same manner as the regular METHOD part.
9.	WRITE_DATATYPE	Not specified. Write a value in the same manner as the regular METHOD part.
10.	READ_REGISTER	Not specified. Write a value in the same manner as the regular METHOD part.
11.	READ_DATA	Not specified. Write a value in the same manner as the regular METHOD part.
12.	READ_DATATYPE	Not specified. Write a value in the same manner as the regular METHOD part.
13.	INTERLOCK	Not specified. Write a value in the same manner as the regular METHOD part.
14.	REQ_FLAG	Not specified. Write a value in the same manner as the regular METHOD part.
15.	END_CONDITION	Not specified. Write a value in the same manner as the regular METHOD part.
16.	ERR_CONDITION	Not specified. Write a value in the same manner as the regular METHOD part.
17.	ERR_REGISTER	Not specified. Write a value in the same manner as the regular METHOD part.
18.	ERR_CODE_RANGE	Not specified. Write a value in the same manner as the regular METHOD part.
19.	RELATED_METHOD	Not specified. Write a value in the same manner as the regular METHOD part.
20.	COMMENT	Not specified. Write a value in the same manner as the regular METHOD part.

(a) Item description example of P_GetEnergyInfoMethod element

```
<p:methodMember label="P_GetEnergyInfoMethod">
  <p:label2><p:item>GetEnergyInfo</p:item></p:label2>
  <p:name><p:item>Get Energy Information</p:item></p:name>
  <p:target><p:item>GetEnergyInfoCommand</p:item></p:target>
  <p:methodType><p:item>COMMAND</p:item></p:methodType>
  <p:writeRegister><p:item>&lt;RWw2&gt;&lt;RWw3&gt;</p:item></p:writeRegister
  >
  <p:writeData><p:item>&lt;$(ARGUMENT.P_ElectricPower.ASSIGN)&gt;&lt;$(ARGUME
  NT.P_ElectricEnergy.ASSIGN)&gt;</p:item></p:writeData>
  <p:writeDatatype><p:item>&lt;WORD&gt;&lt;WORD&gt;</p:item></p:writeDatatype
  >
  <p:readRegister><p:item>&lt;RWr2&gt;&lt;RWr3&gt;</p:item></p:readRegister >
  <p:readData><p:item>&lt;$(ARGUMENT.P_ElectricPower.VALUE)&gt;&lt;$(ARGUMENT
  .P_ElectricEnergy.VALUE)&gt;</p:item></p:readData>
  <p:readDatatype><p:item>&lt;$(ARGUMENT.P_ElectricPower.DATATYPE)&gt;&lt;$(A
  RGUMENT.P_ElectricEnergy.DATATYPE)</p:item></p:readDatatype>
  <p:interlock><p:item>RXF==OFF&amp;amp;&amp;amp;RYF==OFF</p:item></p:interlock>
  <p:reqFlag><p:item>RYF==ON</p:item></p:reqFlag>
  <p:endCondition><p:item>RXF==ON</p:item></p:endCondition>
  <p:comment>Get Electric Power and Electric Energy</p:comment>
</p:methodMember>
```

5.5.8.3 Description details of elements described in MESSAGE part

(1) **Item description of P_GetEnergyInfo** element**

Table 5-385 Item Description Specifications of P_GetEnergyInfo Element**

No.	Item Name	Description Details
1.	LABEL	Write "P_GetEnergyInfo**". Write any string, including spaces, in the area marked by the asterisks (**). (Example: P_GetEnergyInfoMessage)
2.	LABEL2	Not specified. Write a value in the same manner as the regular MESSAGE part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular MESSAGE part.
4.	NAME	Not specified. Write a value in the same manner as the regular MESSAGE part.
5.	TARGET	Write a reference to the COMM_IF_COMMAND part specified by Section 5.5.8.4.
6.	MESSAGE_TYPE	Write "COMMAND".
7.	REQUEST_TYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
8.	REQUEST_DATA	Not specified. Write a value in the same manner as the regular MESSAGE part.
9.	REQUEST_DATATYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
10.	RESPONSE_TYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
11.	RESPONSE_DATA	Not specified. Write a value in the same manner as the regular MESSAGE part.
12.	RESPONSE_DATATYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
13.	ERR_TYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
14.	ERR_CODE_RANGE	Not specified. Write a value in the same manner as the regular MESSAGE part.
15.	RELATED_MESSAGE	Not specified. Write a value in the same manner as the regular MESSAGE part.
16.	COMMENT	Not specified. Write a value in the same manner as the regular MESSAGE part.

(a) Item description example of P_GetEnergyInfoMessage element

```
<p:messageMember label="P_GetEnergyInfoMessage">
  <p:label2><p:item>GetEnergyInfo</p:item></p:label2>
  <p:name><p:item>Get Energy Information</p:item></p:name>
  <p:target><p:item>GetEnergyInfoCommand</p:item></p:target>
  <p:messageType><p:item>COMMAND</p:item></p:messageType>
  <p:requestType><p:item>rdReqST_Binary</p:item></p:requestType>
  <p:requestData>
  <p:item>&lt;0x0613&gt;&lt;0x0000&gt;&lt;0x00000102&gt;&lt;$0x0002&gt;</p:it
  em>
  </p:requestData>
  <p:requestDatatype>
  <p:item>&lt;WORD&gt;&lt;WORD&gt;&lt;DWORD&gt;&lt;WORD&gt;</p:item>
  </p:requestDatatype>
  <p:responseType><p:item>rdResST_Binary</p:item></p:responseType>
  <p:responseData><p:item>&lt;$(ARGUMENT.P_ElectricPower.VALUE)&gt;&lt;$(ARGU
  MENT.P_ElectricEnergy.VALUE)&gt;</p:item></p:responseData>
  <p:responseDatatype><p:item>&lt;$(ARGUMENT.P_ElectricPower.DATATYPE)&gt;&lt
  ;$(ARGUMENT.P_ElectricEnergy.DATATYPE)&gt;</p:item></p:responseDatatype>
  <p:comment>Get Electric Power and Electric Energy</p:comment>
</p:messageMember>
```

5.5.8.4 Description details of elements described in COMM_IF_COMMAND part

(1) **Item description of GetEnergyInfoCommand** element**

Table 5-386 Item Description of GetEnergyInfoCommand Element

No.	Item Name	Description Details
1.	LABEL	Write "GetEnergyInfoCommand**". Write any string, including spaces, in the area marked by the asterisks (**). (Example: GetEnergyInfoCommand1)
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.
4.	NAME	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.
5.	ARGUMENT	[When energy management information is managed on module main body side] Write prohibited. [When energy management information is managed on communication module side] Write a reference to the COMMAND_ARGUMENT part specified by Table 5-383.
6.	REF	[When energy management information is managed on module main body side] Write a reference to the elements of the BLOCK_COMMAND part specified by Section 5.5.8.5. [When energy management information is managed on communication module side] Write prohibited.
7.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.

(a) Item description example of GetEnergyInfoCommand element**

```
<p:commIfCommandMember label="GetEnergyInfoCommand">
  <p:label2><p:item>GetEnergyInfo</p:item></p:label2>
  <p:name><p:item>Get Energy Information</p:item></p:name>
  <p:ref><p:item>BlockSec.BlockCommand.GetEnergyInfoCommand</p:item></p:ref>
</p:commIfCommandMember>
```

5.5.8.5 Description details of elements described in BLOCK_COMMAND part

(1) Item description of GetEnergyInfoCommand** element

Table 5-387 Item Description Specifications of GetEnergyInfoCommand Element**

No.	Item Name	Description Details
1.	LABEL	Write "GetEnergyInfoCommand**". Write any string, including spaces, in the area marked by the asterisks (**). (Example: GetEnergyInfoCommand1)
2.	LABEL2	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.
4.	NAME	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.
5.	ARGUMENT	Write a reference to the COMMAND_ARGUMENT part specified in Table 5-383.
6.	COMMENT	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.

(a) Item description example of GetEnergyInfoCommand element**

```
<p:blockCommandMember label="GetEnergyInfoCommand">
  <p:label2><p:item>GetEnergyInfo</p:item></p:label2>
  <p:name><p:item>Get Energy Information</p:item></p:name>
  <p:argument><p:item>GetEnergyInfoArgument</p:item></p:argument>
</p:blockCommandMember>
```

5.5.8.6 Description details of elements described in COMMAND_ARGUMENT part

(1) Item description of P_AverageCurrent element

Table 5-388 Item Description Specifications of P_AverageCurrent Element

No.	Item Name	Description Details
1.	LABEL	Write "P_AverageCurrent".
2.	LABEL2	Write "P_AverageCurrent" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "Total Current" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_AverageCurrent element

```
<p:commandArgumentMember label="P_AverageCurrent">
  <p:label2><p:item>P_AverageCurrent</p:item></p:label2>
  <p:name><p:item>Total current</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(2) **Item description of P_AverageVoltage element**

Table 5-389 Item Description Specifications of P_AverageVoltage Element

No.	Item Name	Description Details
1.	LABEL	Write "P_AverageVoltage".
2.	LABEL2	Write "P_Average Voltage" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "Total Line Voltage" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_AverageVoltage element

```
<p:commandArgumentMember label="P_AverageVoltage">
  <p:label2><p:item>P_AverageVoltage</p:item></p:label2>
  <p:name><p:item>Total Line Voltage</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(3) **Item description of P_ElectricPower element**

Table 5-390 Item Description Specifications of P_ElectricPower Element

No.	Item Name	Description Details
1.	LABEL	Write "P_ElectricPower".
2.	LABEL2	Write "P_ElectricPower" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "Electric Power" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_ElectricPower element

```
<p:commandArgumentMember label="P_ElectricPower">
  <p:label2><p:item>P_ElectricPower</p:item></p:label2>
  <p:name><p:item>Electric Power</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(4) **Item description of P_ElectricEnergy element**

Table 5-391 Item Description Specifications of P_ElectricEnergy Element

No.	Item Name	Description Details
1.	LABEL	Write "P_ElectricEnergy".
2.	LABEL2	Write "P_ElectricEnergy" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "Electric Energy" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_ElectricEnergy element

```
<p:commandArgumentMember label="P_ElectricEnergy">
  <p:label2><p:item>P_ElectricEnergy</p:item></p:label2>
  <p:name><p:item>Electric Energy</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(5) **Item description of P_PowerDistrubutionTime element**

Table 5-392 Item Description Specifications of P_PowerDistributionTime Element

No.	Item Name	Description Details
1.	LABEL	Write "P_PowerDistributionTime".
2.	LABEL2	Write "P_PowerDistributionTime" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "Integrated Energization Time" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_PowerDistributionTime element

```
<p:commandArgumentMember label="P_PowerDistributionTime">
  <p:label2><p:item>P_PowerDistributionTime</p:item></p:label2>
  <p:name><p:item>Integrated Energization Time</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(6) **Item description of P_RunningTime element**

Table 5-393 Item Description Specifications of P_RunningTime Element

No.	Item Name	Description Details
1.	LABEL	Write "P_RunningTime".
2.	LABEL2	Write "P_RunningTime" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "Actual Operating Time" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_RunningTime element

```
<p:commandArgumentMember label="P_RunningTime">
  <p:label2><p:item>P_RunningTime</p:item></p:label2>
  <p:name><p:item>Actual Operating Time</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(7) **Item description of P_1_PhaseCurrent element**

Table 5-394 Item Description Specifications of P_1_PhaseCurrent Element

No.	Item Name	Description Details
1.	LABEL	Write "P_1_PhaseCurrent".
2.	LABEL2	Write "P_1_PhaseCurrent" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "One-Phase Current" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_1_PhaseCurrent element

```
<p:commandArgumentMember label="P_1_PhaseCurrent">
  <p:label2><p:item>P_1_PhaseCurrent</p:item></p:label2>
  <p:name><p:item>One-Phase Current</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(8) **Item description of P_2_PhaseCurrent element**

Table 5-395 Item Description Specifications of P_2_PhaseCurrent Element

No.	Item Name	Description Details
1.	LABEL	Write "P_2_PhaseCurrent".
2.	LABEL2	Write "P_2_PhaseCurrent" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "Two-Phase Current" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_2_PhaseCurrent element

```
<p:commandArgumentMember label="P_2_PhaseCurrent">
  <p:label2><p:item>P_2_PhaseCurrent</p:item></p:label2>
  <p:name><p:item>Two-PhaseCurrent</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(9) **Item description of P_3_PhaseCurrent element**

Table 5-396 Item Description Specifications of P_3_PhaseCurrent Element

No.	Item Name	Description Details
1.	LABEL	Write "P_3_PhaseCurrent".
2.	LABEL2	Write "P_3_PhaseCurrent" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "Three-Phase Current" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_3_PhaseCurrent element

```
<p:commandArgumentMember label="P_3_PhaseCurrent">
  <p:label2><p:item>P_3_PhaseCurrent</p:item></p:label2>
  <p:name><p:item>Three-Phase Current</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(10) **Item description of P_1_2lineVoltage element**

Table 5-397 Item Description Specifications of P_1_2lineVoltage Element

No.	Item Name	Description Details
1.	LABEL	Write "P_1_2lineVoltage".
2.	LABEL2	Write "P_1_2lineVoltage" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "1-2 Line Voltage" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_1_2lineVoltage element

```
<p:commandArgumentMember label="P_1_2lineVoltage">
  <p:label2><p:item>P_1_2lineVoltage</p:item></p:label2>
  <p:name><p:item>1-2 Line Voltage</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(11) Item description of P_2_3lineVoltage element

Table 5-398 Item Description Specifications of P_2_3lineVoltage Element

No.	Item Name	Description Details
1.	LABEL	Write "P_2_3lineVoltage".
2.	LABEL2	Write "P_2_3lineVoltage" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "2-3 Line Voltage" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_2_3lineVoltage element

```
<p:commandArgumentMember label="P_2_3lineVoltage">
  <p:label2><p:item>P_2_3lineVoltage</p:item></p:label2>
  <p:name><p:item>2-3 Line Voltage</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(12) Item description of P_3_1lineVoltage element

Table 5-399 Item Description Specifications of P_3_1lineVoltage Element

No.	Item Name	Description Details
1.	LABEL	Write "P_3_1lineVoltage".
2.	LABEL2	Write "P_3_1lineVoltage" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "3-1 Line Voltage" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_3_1lineVoltage element

```
<p:commandArgumentMember label="P_3_1lineVoltage">
  <p:label2><p:item>P_3_1lineVoltage</p:item></p:label2>
  <p:name><p:item>3-1 Line Voltage</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(13) Item description of P_1_PhaseCurrentDemand element

Table 5-400 Item Description Specifications of P_1_PhaseCurrentDemand Element

No.	Item Name	Description Details
1.	LABEL	Write "P_1_PhaseCurrentDemand".
2.	LABEL2	Write "P_1_PhaseCurrentDemand" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "One-Phase Demand Current" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_1_PhaseCurrentDemand element

```
<p:commandArgumentMember label="P_1_PhaseCurrentDemand">
  <p:label2><p:item>P_1_PhaseCurrentDemand</p:item></p:label2>
  <p:name><p:item>One-Phase Demand Current</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(14) Item description of P_2_PhaseCurrentDemand element

Table 5-401 Item Description Specifications of P_2_PhaseCurrentDemand Element

No.	Item Name	Description Details
1.	LABEL	Write "P_2_PhaseCurrentDemand".
2.	LABEL2	Write "P_2_PhaseCurrentDemand" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "Two-Phase Demand Current" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_2_PhaseCurrentDemand element

```
<p:commandArgumentMember label="P_2_PhaseCurrentDemand">
  <p:label2><p:item>P_2_PhaseCurrentDemand</p:item></p:label2>
  <p:name><p:item>Two-Phase Demand Current</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(15) Item description of P_3_PhaseCurrentDemand element

Table 5-402 Item Description Specifications of P_3_PhaseCurrentDemand Element

No.	Item Name	Description Details
1.	LABEL	Write "P_3_PhaseCurrentDemand".
2.	LABEL2	Write "P_3_PhaseCurrentDemand" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "Three-Phase Demand Current" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_3_PhaseCurrentDemand element

```
<p:commandArgumentMember label="P_3_PhaseCurrentDemand">
  <p:label2><p:item>P_3_PhaseCurrentDemand</p:item></p:label2>
  <p:name><p:item>Three-Phase Demand Current</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(16) **Item description of P_ElectricPowerDemand element**

Table 5-403 Item Description Specifications of P_ElectricPowerDemand Element

No.	Item Name	Description Details
1.	LABEL	Write "P_ElectricPowerDemand".
2.	LABEL2	Write "P_ElectricPowerDemand" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "Demand Electric Power" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_ElectricPowerDemand element

```
<p:commandArgumentMember label="P_ElectricPowerDemand">
  <p:label2><p:item>P_ElectricPowerDemand</p:item></p:label2>
  <p:name><p:item>Demand Electric Power</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(17) Item description of P_PowerFactor element

Table 5-404 Item Description Specifications of P_PowerFactor Element

No.	Item Name	Description Details
1.	LABEL	Write "P_PowerFactor".
2.	LABEL2	Write "P_PowerFactor" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "Power Factor" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_PowerFactor element

```
<p:commandArgumentMember label="P_PowerFactor">
  <p:label2><p:item>P_PowerFactor</p:item></p:label2>
  <p:name><p:item>Power Factor</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(18) Item description of P_Frequency element

Table 5-405 Item Description Specifications of P_Frequency Element

No.	Item Name	Description Details
1.	LABEL	Write "P_Frequency".
2.	LABEL2	Write "P_Frequency" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "Frequency" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_Frequency element

```
<p:commandArgumentMember label="P_Frequency">
  <p:label2><p:item>P_Frequency</p:item></p:label2>
  <p:name><p:item>Frequency</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(19) Item description of P_ReactiveEnergy element

Table 5-406 Item Description Specifications of P_ReactiveEnergy Element

No.	Item Name	Description Details
1.	LABEL	Write "P_ReactiveEnergy".
2.	LABEL2	Write "P_ReactiveEnergy" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "Reactive Energy" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_ReactiveEnergy element

```
<p:commandArgumentMember label="P_ReactiveEnergy">
  <p:label2><p:item>P_ReactiveEnergy</p:item></p:label2>
  <p:name><p:item>Reactive Energy</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(20) **Item description of P_1_0PhaseVoltage element**

Table 5-407 Item Description Specifications of P_1_0PhaseVoltage Element

No.	Item Name	Description Details
1.	LABEL	Write "P_1_0PhaseVoltage".
2.	LABEL2	Write "P_1_0PhaseVoltage" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "1-0 Phase Voltage" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_1_0PhaseVoltage element

```
<p:commandArgumentMember label="P_1_0PhaseVoltage">
  <p:label2><p:item>P_1_0PhaseVoltage</p:item></p:label2>
  <p:name><p:item>One-Phase Voltage</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(21) **Item description of P_2_0PhaseVoltage element**

Table 5-408 Item Description Specifications of P_2_0PhaseVoltage Element

No.	Item Name	Description Details
1.	LABEL	Write "P_2_0PhaseVoltage".
2.	LABEL2	Write "P_2_0PhaseVoltage" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "2-0 Phase Voltage" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_2_0PhaseVoltage element

```
<p:commandArgumentMember label="P_2_0PhaseVoltage">
  <p:label2><p:item>P_2_0PhaseVoltage</p:item></p:label2>
  <p:name><p:item>Two-Phase Voltage</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

(22) **Item description of P_3_0PhaseVoltage element**

Table 5-409 Item Description Specifications of P_3_0PhaseVoltage Element

No.	Item Name	Description Details
1.	LABEL	Write "P_3_0PhaseVoltage".
2.	LABEL2	Write "P_3_0PhaseVoltage" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write "3-0 Phase Voltage" in the case of English CSP+ files. [Recommended specifications]
5.	DATATYPE	Select and write one of the following: unsigned integer type, real number type.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part. Normally, write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of P_3_0PhaseVoltage element

```
<p:commandArgumentMember label="P_3_0PhaseVoltage">
  <p:label2><p:item>P_3_0PhaseVoltage</p:item></p:label2>
  <p:name><p:item>Three-Phase Voltage</p:item></p:name>
  <p:datatype><p:item>UINT16</p:item></p:datatype>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

5.5.9 Convention for acquiring status codes

SLMP defines specifications related to status code acquisition.

5.5.9.1 Overview

This section shows the convention for acquiring status codes from a module using SLMP to display the meaning of the status codes, corrective actions for them, and more. This convention, in the current version, can only be written in CSP+ for the Ethernet connection unit. Figure 5-53 shows an image of the structure when the acquisition of the status code is described.

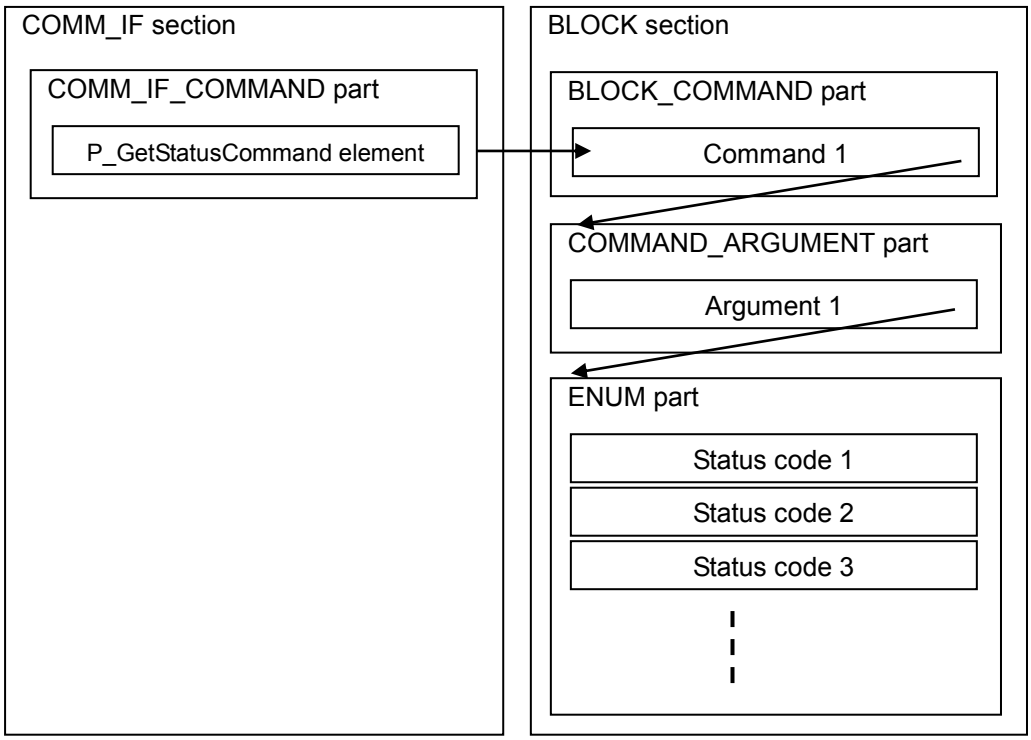


Figure 5-53 Image of Structure When Acquisition of Status Information Is Described

One element is described in each of the COMM_IF_COMMAND part, BLOCK_COMMAND part, and COMMAND_ARGUMENT part.
A status code is acquired with use of the GetStatus command of SLMP.

Table 5-410 lists the elements described in the COMM_IF_COMMAND part. For details of the description specifications of the elements, see Section 5.5.9.2.

Table 5-410 List of Elements Described in COMM_IF_COMMAND Part

No.	Element Name	Description Details	Required/Optional
1.	P_GetStatusCommand	Describes the command information used to acquire the status code.	Required

Table 5-411 lists the elements described in the BLOCK_COMMAND part. For details of the description specifications of the elements, see Section 5.5.9.3.

Table 5-411 List of Elements Described in BLOCK_COMMAND Part

No.	Element Name	Description Details	Required/Optional
1.	(Name Optional)	Describes the command information used to acquire the status code.	Required*1

Table 5-412 lists the elements described in the COMMAND_ARGUMENT part. For details of the description specifications of the elements, see Section 5.5.9.4.

Table 5-412 List of Elements Described in COMMAND_ARGUMENT Part

No.	Element Name	Description Details	Required/Optional
1.	(Name Optional)	Describes the argument of the status code acquisition command (status code).	Required

Table 5-413 lists the elements described in the ENUM part. For details of the description specifications of the elements, see Section 5.5.9.4.

Table 5-413 List of Elements Described in ENUM Part

No.	Element Name	Description Details	Required/Optional
1.	(Name Optional)	Describes a status code value, its meaning, and a corrective action.	Required

Note that while multiple parts are used to describe the acquisition of status codes, there are no regulations regarding the label name of any part. (The CSP+ file creator can freely assign label names.)

5.5.9.2 Description details of elements described in COMM_IF_COMMAND part

(1) Item description of P_GetStatusCommand element

Table 5-414 Item Description of P_GetStatusCommand Element

No.	Item Name	Description Details
1.	LABEL	Write "P_GetStatusCommand".
2.	LABEL2	Write "P_GetStatusCommand" in the case of English CSP+ files. [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.
4.	NAME	Write "Status Code Acquisition Command" in the case of English CSP+ files. [Recommended specifications]
5.	ARGUMENT	Write prohibited.
6.	REF	Write a reference to the BLOCK_COMMAND part specified by Section 5.5.9.3.
7.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.

(a) Item description example of P_GetStatusCommand element

```
<p:commIfCommandMember label="P_GetStatusCommand">
  <p:label2><p:item>P_GetStatusCommand</p:item></p:label2>
  <p:name><p:item>Status Code Acquisition Command</p:item></p:name>
  <p:ref><p:item>BlockSec.BlockCommand.GetStatusCommand</p:item></p:ref>
</p:commIfCommandMember>
```

5.5.9.3 Description details of elements described in BLOCK_COMMAND part

(1) Item description of BLOCK_COMMAND part element

Table 5-415 Item Description Specifications of Element

No.	Item Name	Description Details
1.	LABEL	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.
2.	LABEL2	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.
4.	NAME	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.
5.	ARGUMENT	Write a reference to the COMMAND_ARGUMENT part specified by Section 5.5.9.4.
6.	COMMENT	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.

(a) Item description example of element

```
<p:blockCommandMember label="GetStatusCommand">
  <p:label2><p:item>P_GetStatusCommand</p:item></p:label2>
  <p:name><p:item>Status Code Acquisition Command</p:item></p:name>
  <p:argument><p:item>StatusArgument</p:item></p:argument>
</p:blockCommandMember>
```

5.5.9.4 Description details of elements described in COMMAND_ARGUMENT part

(1) Item description of COMMAND_ARGUMENT part element

Table 5-416 Item Description Specifications of Element

No.	Item Name	Description Details
1.	LABEL	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
5.	DATATYPE	Write "WORD".
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Write a reference to the COMMAND_ARGUMENT part specified by RANGE.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Write "R".
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of ErrorCode element

```
<p:commandArgumentMember label="StatusCode">
  <p:label2><p:item>StatusCode</p:item></p:label2>
  <p:name><p:item>Status Code</p:item></p:name>
  <p:datatype><p:item>WORD</p:item></p:datatype>
  <p:range><p:enumRefItem>EnumStatusCode</p:enumRefItem></p:range>
  <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

5.5.9.5 Description details of elements described in ENUM part

(1) Item description of ENUM part element

Table 5-417 Item Description Specifications of ENUM Part Element

No.	Item Name	Description Details
1.	LABEL	Not specified. Write a value in the same manner as the regular ENUM part.
2.	LABEL2	Not specified. Write a value in the same manner as the regular ENUM part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular ENUM part.
4.	NAME	Describes the name of a status code value.
5.	CODE	Describes a status code value.
6.	COMMENT	Describes the meaning, a corrective action, and more of the status code.

(a) Item description example of ENUM part element

```
<p:enum label="EnumStatusCode ">
  <p:enumMember label="enum1">
    <p:name><p:item>Hardware Error</p:item></p:name>
    <p:code><p:item>1</p:item></p:code>
    <p:comment><p:item>Turn off the power once, and turn on the power again.
    </p:item></p:comment>
  </p:enumMember>
  <p:enumMember label="enum2">
    <p:name><p:item>User Range Read Error</p:item></p:name>
    <p:code><p:item>2</p:item></p:code>
    <p:comment><p:item>This item turns on when an error is present in the user
    range read from a flash memory. Turning on the error reset request flag can
    reset (turn off) this flag. </p:item></p:comment>
  </p:enumMember>
  <p:enumMember label="enum3">
    <p:name><p:item>Flash Memory Write Error</p:item></p:name>
    <p:code><p:item>3</p:item></p:code>
    <p:comment><p:item>This item turns on when writing to a flash memory failed.
    When this flag turns on, reset cannot be performed using the error reset request
    flag because of a failure of the module itself. </p:item></p:comment>
  </p:enumMember>
</p:enum>
```

5.5.10 Specifications related to CC-Link IE TSN

5.5.10.1 Parts and elements related to error related information

The module has a function that records the information of any errors that occur during module operation, and acquires that error information as related information via SNMP from an external source. This section describes the specifications of parts and elements when writing error related information acquisition processing.

5.5.10.1.1. Overview

Write error related information acquisition in the MESSAGE, COMM_IF_COMMAND, BLOCK_COMMAND, COMMAND_ARGUMENT, and ENUM parts. Figure 5-54 shows an image of the structure when describing the error related information acquisition. (The specifications are only for CC-Link IE TSN since the RELATED_ELE item of the ENUM part is used.)

LABEL2 and NAME are recommended specifications, and other strings can be written.

In the case of devices for which station mode can be changed (profiles have multiple COMM_IF sections), matching descriptions of each COMM_IF section is required. For details, see the description methods of each part.

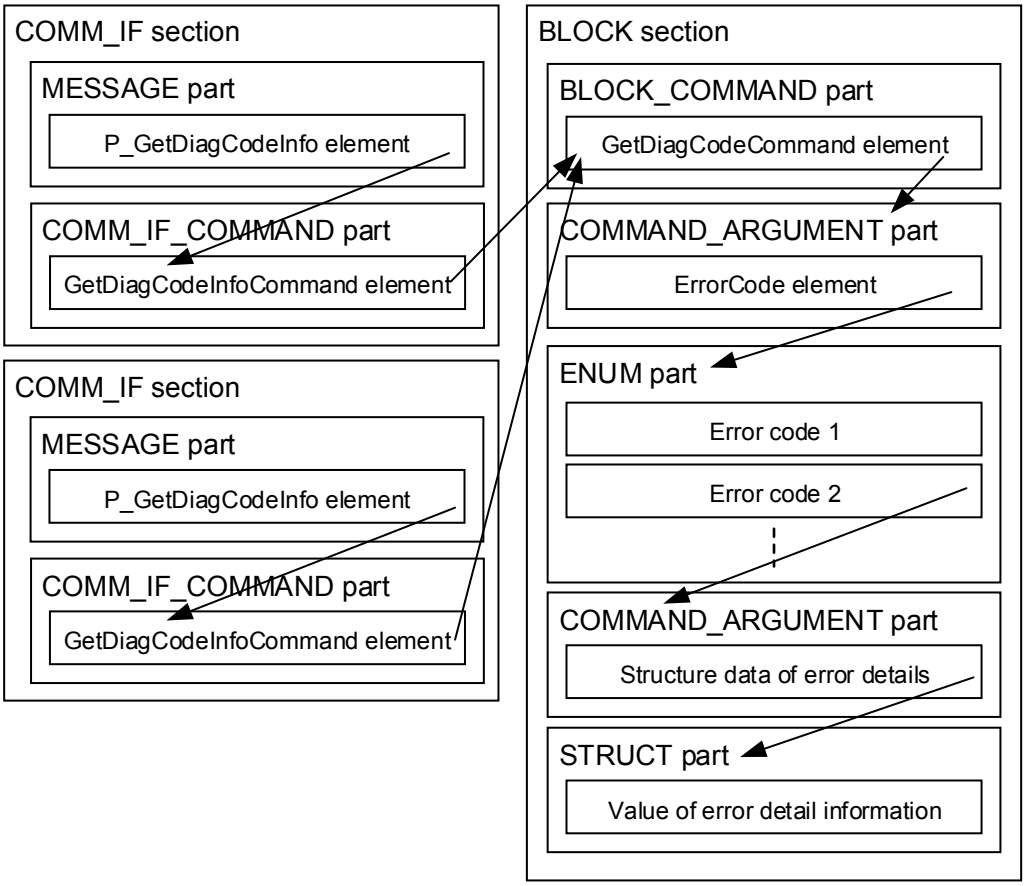


Figure 5-54 Image of Structure When Describing Error Related Information Acquisition

Table 5-418 shows a list of elements described in the MESSAGE part. For details of the description specifications of each element, see Section 5.5.10.1.2.

When a MESSAGE part has been written to one COMM_IF section of the profile which has multiple COMM_IF sections, the MESSAGE part should be written to all the COMM_IF sections.

Table 5-418 List of Elements Written in MESSAGE Part

No.	Element Name	Description Details	Required/Optional
1.	P_GetDiagCodeInfo	Write the information used for diagnostics code information acquisition.	Required

Table 5-419 shows a list of elements described in the COMM_IF_COMMAND part. For details of the description specifications of each element, see Section 5.5.10.1.3.

When the profile has multiple sections, refer to the same BLOCK_COMMAND part element for the error information acquisition from COMM_IF_COMMAND parts for error information acquisition of all the COMM_IF sections.

Table 5-419 List of Elements Written in COMM_IF_COMMAND Part

No.	Element Name	Description Details	Required/Optional
1.	GetDiagCodeInfoCommand	Write the command information used for diagnostics code information acquisition.	Required

Table 5-420 shows a list of elements described in the BLOCK_COMMAND part. For details of the description specifications of each element, see Section 5.5.10.1.4.

A single element of BLOCK_COMMAND part for the error related information acquisition can be written in a profile.

Table 5-420 List of Elements Written in BLOCK_COMMAND Part

No.	Element Name	Description Details	Required/Optional
1.	GetDiagCodeCommand	Write the command information used for diagnostics code information acquisition.	Required

Table 5-421 shows a list of elements described in the COMMAND_ARGUMENT part. For details of the description specifications of each element, see Section 5.5.10.1.5.

Table 5-421 List of Elements Written in COMMAND_ARGUMENT Part

No.	Element Name	Description Details	Required/Optional
1.	ErrorCode	Write the error code information.	Required

Table 5-422 shows a list of elements described in the ENUM part. For details of the description specifications of each element, see Section 5.5.10.1.6.

Table 5-422 List of Elements Written in ENUM Part

No.	Element Name	Description Details	Required/Optional
1.	(Any name)	Write the error detailed information.	Required

Table 5-423 shows a list of elements described in the COMMAND_ARGUMENT part for the detailed information structure definition. Describe the data structure for the detailed information according to the following specifications.

[1] From the error code to the error details size are not described.

[2] Describe the elements of COMMAND_ARGUMENT in order so that the order of the data from the error details size is indicated.

[3] To divide one word so that multiple information is held, express the information with a STRUCT type. Please note that data larger than one word cannot be expressed with one STRUCT. (Use is possible for only the division of one word.)

[4] The STRING(x) type can be used as the DATATYPE item. The following shows the precautions on

the specifications of the error detailed data returned by a device when using STRING(x).

• Control characters excluding null characters cannot be inserted in the error details data.

• An error detailed data area for the number of characters specified in DATATYPE must be allocated. For example, when STRING(16) is used, an error detailed data area for 8 words must be allocated even for the case when 5 characters (3 words) are returned (do not use the 3 words at the end for other error details data).

[5] For the specifiable data specifications (data size and others) as detailed information, refer to "CC-Link IE TSN Specifications (Overview) (BAP-C2011ENG-001).
Describing the element is optional. For details of the description specifications of each element, see Section 5.5.10.1.7.

Table 5-423 List of Elements Written in COMMAND_ARGUMENT Part for Detailed Information Structure Definition

No.	Element Name	Description Details	Required/Optional
1.	(Any name)	Write the structure of the error detailed information.	Optional

Table 5-424 shows a list of elements described in the STRUCT part. For details of the description specifications of each element, see Section 5.5.10.1.8.

Table 5-424 List of Elements Written in STRUCT Part

No.	Element Name	Description Details	Required/Optional
1.	(Any name)	Write the structure for dividing the error details information.	Optional

5.5.10.1.2. Description details of elements written in MESSAGE part

(1) Item description of P_GetDiagCodeInfo element

Table 5-425 Item Description Specifications of P_GetDiagCodeInfo Element

No.	Item Name	Description Details
1.	LABEL	Write "P_GetDiagCodeInfo".
2.	LABEL2	Write "P_GetDiagCodeInfo". [Recommended specifications]
3.	CATEGORY	Write prohibited.
4.	NAME	Write "Diagnostics Code Information". [Recommended specifications]
5.	TARGET	Write a reference to the COMM_IF_COMMAND part that you will write in Section 5.5.10.1.3.
6.	MESSAGE_TYPE	Write "OTHER".
7.	REQUEST_TYPE	Write prohibited.
8.	REQUEST_DATA	Write prohibited.
9.	REQUEST_DATATYPE	Write prohibited.
10.	RESPONSE_TYPE	Write prohibited.
11.	RESPONSE_DATA	Write prohibited.
12.	RESPONSE_DATATYPE	Write prohibited.
13.	ERR_TYPE	Write prohibited.
14.	ERR_CODE_RANGE	Write prohibited.
15.	RELATED_MESSAGE	Write prohibited.
16.	COMMENT	Write prohibited.

(a) Item description example of P_GetDiagCodeInfo element

```
<p:messageMember label="P_GetDiagCodeInfo">
   <p:label2><p:item>P_GetDiagCodeInfo</p:item></p:label2>
   <p:name><p:item>Diagnostics Code Information</p:item></p:name>
   <p:target><p:item>CommandSec.GetDiagCodeInfoCommand</p:item></p:target>
   <p:messageType><p:item>OTHER</p:item></p:messageType>
</p:messageMember>
```

5.5.10.1.3. Description details of elements written in COMM_IF_COMMAND part

(1) Item description of GetDiagCodeInfoCommand element

Table 5-426 Item Description Specifications of GetDiagCodeInfoCommand Element

No.	Item Name	Description Details
1.	LABEL	Write "GetDiagCodeInfoCommand".
2.	LABEL2	Write prohibited.
3.	CATEGORY	Write prohibited.
4.	NAME	Write prohibited.
5.	ARGUMENT	Write prohibited.
6.	REF	Write a reference to the BLOCK_COMMAND part that you will write in Section 5.5.10.1.4.
7.	COMMENT	Write prohibited.

(a) Item description example of GetDiagCodeInfoCommand element

```
<p:commIfCommandMember label="GetDiagCodeInfoCommand">
   <p:ref><p:item>BlockSec.CommandSec.GetDiagCodeCommand</p:item></p:ref>
</p:commIfCommandMember>
```

5.5.10.1.4. Description details of elements written in BLOCK_COMMAND part

(1) Item description of GetDiagCodeCommand element

Table 5-427 Item Description Specifications of GetDiagCodeCommand Element

No.	Item Name	Description Details
1.	LABEL	Write "GetDiagCodeCommand".
2.	LABEL2	Write prohibited.
3.	CATEGORY	Write prohibited.
4.	NAME	Write "Diagnostics Code Information Acquisition Command". [Recommended specifications]
5.	ARGUMENT	Write a reference to the COMMAND_ARGUMENT part that you will write in Section 5.5.10.1.5.
6.	COMMENT	Write prohibited.

(a) Item description example of GetErrorLogCommand element

```
<p:blockCommandMember label="GetErrorLogCommand">
   <p:label2><p:item>GetErrorLogCommand</p:item></p:label2>
   <p:name><p:item>Diagnostics Code Information Acquisition
   Command</p:item></p:name>
   <p:argument><p:item>Command_Arg</p:item></p:argument>
</p:blockCommandMember>
```

5.5.10.1.5. Description details of elements written in COMMAND_ARGUMENT part

(1) Item description of ErrorCode element

Table 5-428 Item Description Specifications of ErrorCode Element

No.	Item Name	Description Details
1.	LABEL	Write "ErrorCode".
2.	LABEL2	Write prohibited.
3.	CATEGORY	Write prohibited.
4.	NAME	Write "Error Code". [Recommended specifications]
5.	DATATYPE	Write "WORD".
6.	DEFAULT	Write prohibited.
7.	RANGE	Write a reference to the ENUM part that you will write in Section 5.5.10.1.6.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "R".
11.	ASSIGN	Write prohibited.
12.	REF	Write prohibited.
13.	COMMENT	Write prohibited.

(a) Item description example of ErrorCode element

```
<p:commandArgumentMember label="ErrorCode">
   <p:label2><p:item>ErrorCode</p:item></p:label2>
   <p:name><p:item>Error Code</p:item></p:name>
   <p:datatype><p:item>WORD</p:item></p:datatype>
   <p:range><p:enumRefItem>ErrorCodeList</p:enumRefItem></p:range>
   <p:access><p:item>R</p:item></p:access>
</p:commandArgumentMember>
```

5.5.10.1.6. Description details of elements written in ENUM part

(1) Item description of ENUM part element

Table 5-429 Item Description Specifications of Elements of ENUM Part

No.	Item Name	Description Details
1.	LABEL	Not specified. Write a value in the same manner as the regular ENUM part.
2.	LABEL2	Not specified. Write a value in the same manner as the regular ENUM part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular ENUM part.
4.	NAME	Write the name (overview) of the error code.
5.	CODE	Write the error code (hexadecimal).
6.	RELATED_ELE	In the case of an error that has error details information, write the reference for the corresponding detailed information structure definition part (COMMAND_ARGUMENT). This can be omitted if the error does not have error details information.
7.	COMMENT1	Write the error cause.
8.	COMMENT2	Write the remedy.

(a) Item description example of ENUM part element

```
<p:enum label="ErrorCodeList">
   <p:enumMember label="enum1">
      <p:name><p:item>Hardware Code</p:item></p:name>
      <p:code><p:item>0x10</p:item></p:code>
      <p:comment index="1"><p:item>Hardware Error of Module</p:item></p:comment>
      <p:comment index="2"><p:item>Turn off and on the module power.
      </p:item></p:comment>
   </p:enumMember>
   <p:enumMember label="enum2">
      <p:name><p:item>Error Outside Input Value Range</p:item></p:name>
      <p:code><p:item>0x105</p:item></p:code>
      <p:relatedEle><p:item>InputError</p:item></p:relatedEle>
      <p:comment index="1"><p:item>A value larger than the maximum value set for the
      parameter has been input.</p:item></p:comment>
      <p:comment index="2"><p:item>Review the input or review the parameter setting.
      </p:item></p:comment>
   </p:enumMember>
   <p:enumMember label="enum3">
      <p:name><p:item>Remote Buffer Memory Access Error</p:item></p:name>
      <p:code><p:item>0x150</p:item></p:code>
      <p:comment index="1"><p:item>Outside the range of the remote buffer memory is
      accessed by the REMFR/REMTO command. </p:item></p:comment>
      <p:comment index="2"><p:item>Correct the setting data of the REMFR/REMTO command
      so that access is performed within the range of the remote buffer memory.
      </p:item></p:comment>
   </p:enumMember>
</p:enum>
```

5.5.10.1.7. Description details of elements written in COMMAND_ARGUMENT part for detailed information structure definition

(1) Item description of COMMAND_ARGUMENT part element for detailed information structure definition

Table 5-430 Item Description Specifications of COMMAND_ARGUMENT Part Element for Detailed Information Structure Definition

No.	Item Name	Description Details
1.	LABEL	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
4.	NAME	Write the name of the data structure definition of the detailed information.
5.	DATATYPE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
11.	ASSIGN	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
12.	REF	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.
13.	COMMENT	Not specified. Write a value in the same manner as the regular COMMAND_ARGUMENT part.

(a) Item description example of COMMAND_ARGUMENT part element

```
<p:commandArgument label="InputError">
   <p:commandArgumentMember label="ele1">
      <p:name><p:item>Axis Number</p:item></p:name>
      <p:datatype><p:item>UINT16</p:item></p:datatype>
   </p:commandArgumentMember>
   <p:commandArgumentMember label="ele2">
      <p:name><p:item>Input Value information</p:item></p:name>
      <p:datatype><p:item>STRUCT InputInfo</p:item></p:datatype>
   </p:commandArgumentMember>
</p:commandArgument>
```

5.5.10.1.8. Description details of elements written in STRUCT part

(1) Item description of STRUCT part element

Table 5-431 Item Description Specifications of STRUCT Part Element

No.	Item Name	Description Details
1.	LABEL	Not specified. Write a value in the same manner as the regular STRUCT part.
2.	LABEL2	Not specified. Write a value in the same manner as the regular STRUCT part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular STRUCT part.
4.	NAME	Not specified. Write a value in the same manner as the regular STRUCT part.
5.	DATATYPE	Not specified. Write a value in the same manner as the regular STRUCT part.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular STRUCT part.
7.	RANGE	Not specified. Write a value in the same manner as the regular STRUCT part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular STRUCT part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular STRUCT part.
10.	OFFSET	Not specified. Write a value in the same manner as the regular STRUCT part.
11.	REF	Not specified. Write a value in the same manner as the regular STRUCT part.
12.	COMMENT	Not specified. Write a value in the same manner as the regular STRUCT part.

(a) Item description example of ErrorCode element

```
<p:struct label="InputInfo">
   <p:structMember label="ele1">
      <p:name><p:item>Current Value</p:item></p:name>
      <p:datatype><p:item>INT8</p:item></p:datatype>
      <p:offset><p:item>0.0</p:item></p:offset>
   </p:structMember>
   <p:structMember label="ele2">
      <p:name><p:item>Maximum Value</p:item></p:name>
      <p:datatype><p:item>INT8</p:item></p:datatype>
      <p:offset><p:item>0.8</p:item></p:offset>
   </p:structMember>
</p:struct>
```

Figure 5-55 shows examples of parts and elements of the error related information.

MESSAGE

	LABEL	LABEL2	NAME	TARGET	MESSAGE_TYPE
1	P_GetDiagCodeInfo	P_GetDiagCodeInfo	Diagnostics Code In	CommandSec.GetDiagCodeInfoCommand	OTHER

COMM_IF_COMMAND (LABEL=CommandSec)

	LABEL	REF
1	GetDiagCodeInfoCommand	BlockSec.CommandSec.GetDiagCodeCommand

BLOCK_COMMAND (LABEL=CommandSec)

	LABEL	NAME	ARGUMENT
1	GetDiagCodeCommand	Diagnostics Code Information A	Command_Arg

COMMAND_ARGUMENT (LABEL=Command_Arg)

	LABEL	NAME	DATATYPE	RANGE	ACCESS
1	ErrorCode	Error Code	WORD	ErrorCodeList	R

ENUM (LABEL=ErrorCodeList)

	LABEL	NAME	CODE	RELATED_ELE	COMMENT1	COMMENT2
1	enum1	Hardware Error	0x10		Hardware Error of Module	Turn off and on the module power. If the error occurs again, the module may be damaged.
2	enum2	Error Outside Input Value Range	0x105	InputError	A value larger than the maximum value set for the parameter has been input.	Review the input or review the parameter setting.
3	enum3	Remote Buffer Memory Access Error	0x150		Outside the range of the remote buffer memory is accessed by the REMFR/REMTO command.	Correct the setting data of the REMFR/REMTO command so that access is performed within the range of the remote buffer memory.

COMMAND_ARGUMENT (LABEL=InputError)

	LABEL	NAME	DATATYPE
1	ele1	Axis Number	UINT16
2	ele2	Input Value in	STRUCT InputInfo

STRUCT (LABEL=InputInfo)

	LABEL	NAME	DATATYPE	OFFSET
1	ele1	Current V	INT8	0.0
2	ele2	Maximu	INT8	0.8

Figure 5-55 Description Example of Parts and Elements Related to Error Related Information

5.5.10.2 Convention related to MESSAGE for slave station parameter automatic setting

With CC-Link IE TSN, the parameters are written automatically when a slave station is reconnected or introduced by turning on its power or online exchange. This section describes how to write slave station parameter automatic setting.

5.5.10.2.1. Overview

Slave station parameter automatic setting is written in the MESSAGE part. The following two conventions are available as description methods.

[1] Method to indicate automatic setting using only one element (method to indicate using No. 1 of Table 5-432)
All of the parameters exist in consecutive address areas, and description with this method is possible when batch automatic setting is possible.

[2] Method to write "SEQ_TARGET" in TARGET to indicate automatic setting using multiple elements (method to refer to multiple No. 3 elements from No. 2 of Table 5-432)
When the parameters exist in non-consecutive address areas, write the No. 3 element of Table 5-432 for the automatic setting process of every area.

Only any one element can be written for one COMM_IF section. Figure 5-56 shows an image of the structure of slave station parameter automatic setting.

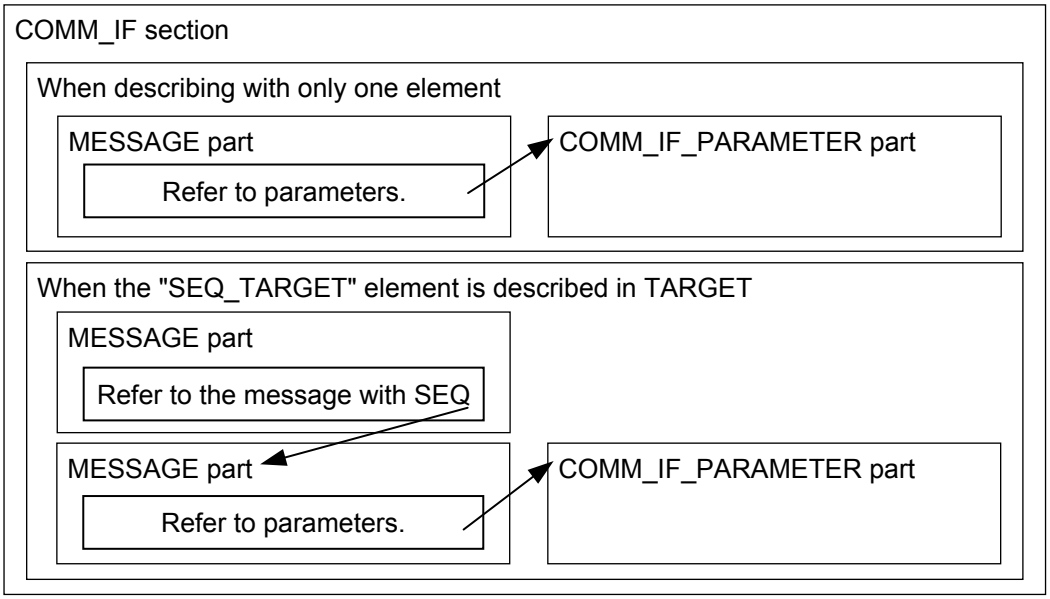


Figure 5-56 Image of Structure When Describing MESSAGE for Slave Station Parameter Automatic Setting

Table 5-432 shows a list of elements described in the MESSAGE part. For details of the description specifications of each element, see Section 5.5.10.2.2.

Table 5-432 List of Elements Written in MESSAGE Part

No.	Element Name	Description Details	Required/Optional
1.	(Any name)	Write a message in which only one element (reference of COMM_IF_PARAMETER part to TARGET) is written.	Required
2.	(Any name)	Write a message in which "SEQ_TARGET" is written in TARGET.	Required
3.	(Any name)	Write a message of an element referred to from SEQ_TARGET of No. 2.	Required

Table 5-433 shows a list of elements described in the COMM_IF_PARAMETER part. For details of the description specifications of each element, see Section 5.5.10.2.3.

Table 5-433 List of Elements Written in COMM_IF_PARAMETER Part

No.	Element Name	Description Details	Required/Optional
1.	(Any name)	Write the parameter information to set in slave station parameter automatic setting.	Required

5.5.10.2.2. Description details of elements written in MESSAGE part

(1) Item description of element to write using only one element

Table 5-434 Item Description Specifications of Element Written Using Only One Element

No.	Item Name	Description Details
1.	LABEL	Not specified. Write a value in the same manner as the regular MESSAGE part. The prefix cannot be set to "SLMP" because this is not an SLMP-based MESSAGE.
2.	LABEL2	Not specified. Write a value in the same manner as the regular MESSAGE part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular MESSAGE part.
4.	NAME	Not specified. Write a value in the same manner as the regular MESSAGE part.
5.	TARGET	Write the reference to the COMM_IF_PARAMETER part that you will write in Section 5.5.10.2.3 in the format of "'Part Name'.*". *1
6.	MESSAGE_TYPE	Write "AUTO_PARAMETER".
7.	REQUEST_TYPE	Write prohibited.
8.	REQUEST_DATA	Write the three items of the parameter starting address (hexadecimal), target parameter size (hexadecimal), and "$(*.VALUE)" using the <> convention.
9.	REQUEST_DATATYPE	<DWORD><WORD><$(*.DATATYPE)>
10.	RESPONSE_TYPE	Write prohibited.
11.	RESPONSE_DATA	Write prohibited.
12.	RESPONSE_DATATYPE	Write prohibited.
13.	ERR_TYPE	Write prohibited.
14.	ERR_CODE_RANGE	Write prohibited.
15.	RELATED_MESSAGE	Write prohibited.
16.	COMMENT	Not specified. Write a value in the same manner as the regular MESSAGE part.

*1: All elements of COMM_IF_PARAMETER (all elements specified as TARGET of MESSAGE for the process in which TARGET is written with PARAMETER) that are the parameter writing targets need to be written so that they become the targets of parameter automatic setting. However, elements whose DATATYPE is constant (CONST) are out of this restriction. Therefore, some of the elements can be written only in either part.

(a) Item description example of element

```
<p:messageMember label="P_WriteParam">
   <p:label2><p:item>P_WriteParam</p:item></p:label2>
   <p:name><p:item>Parameter Batch Write</p:item></p:name>
   <p:target><p:item>WriteParam.*</p:item></p:target>
   <p:messageType><p:item>AUTO_PARAMETER</p:item></p:messageType>
   <p:requestData>
      <p:item>item>&lt;0x0401&gt;&lt;0x0000&gt;&lt;0x00000000&gt;&lt;$0x0006&gt;</
      p:item>
   </p:requestData>
   <p:requestDatatype>
      <p:item>&lt;WORD&gt;&lt;WORD&gt;&lt;DWORD&gt;&lt;WORD&gt;</p:item>
   </p:requestDatatype>
</p:messageMember>
```

(2) Item description of element to write "SEQ_TARGET" to TARGET

Table 5-435 Item Description Specifications of Element to Write "SEQ_TARGET" to TARGET

No.	Item Name	Description Details
1.	LABEL	Not specified. Write a value in the same manner as the regular MESSAGE part. The prefix cannot be set to "SLMP" because this is not an SLMP-based MESSAGE.
2.	LABEL2	Not specified. Write a value in the same manner as the regular MESSAGE part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular MESSAGE part.
4.	NAME	Not specified. Write a value in the same manner as the regular MESSAGE part.
5.	TARGET	Write "SEQ_TARGET".
6.	MESSAGE_TYPE	Write "AUTO_PARAMETER".
7.	REQUEST_TYPE	Write prohibited.
8.	REQUEST_DATA	Write prohibited.
9.	REQUEST_DATATYPE	Write prohibited.
10.	RESPONSE_TYPE	Write prohibited.
11.	RESPONSE_DATA	Write prohibited.
12.	RESPONSE_DATATYPE	Write prohibited.
13.	ERR_TYPE	Write prohibited.
14.	ERR_CODE_RANGE	Write prohibited.
15.	RELATED_MESSAGE	Refer to related elements from "SEQ".
16.	COMMENT	Not specified. Write a value in the same manner as the regular MESSAGE part.

(a) Item description example of element

```
<p:messageMember label="P_SEQMessage">
   <p:label2><p:item>P_SEQMessage</p:item></p:label2>
   <p:name><p:item>SEQ Batch Write</p:item></p:name>
   <p:target><p:item>SEQ_TARGET</p:item></p:target>
   <p:messageType><p:item>AUTO_PARAMETER</p:item></p:messageType>
   <p:relatedMessage>
      <p:item>&lt;SEQ MessageA&gt;&lt;SEQ MessageB&gt;</p:item>
   </p:relatedMessage>
</p:messageMember>
```

(3) Item description of elements referenced from the elements of "SEQ_TARGET"

Table 5-436 Item Description Specifications of Elements Referenced from "SEQ_TARGET" Element

No.	Item Name	Description Details
1.	LABEL	Not specified. Write a value in the same manner as the regular MESSAGE part. The prefix cannot be set to "SLMP" because this is not an SLMP-based MESSAGE.
2.	LABEL2	Not specified. Write a value in the same manner as the regular MESSAGE part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular MESSAGE part.
4.	NAME	Not specified. Write a value in the same manner as the regular MESSAGE part.
5.	TARGET	Write the reference to the COMM_IF_PARAMETER part that you will write in Section 5.5.10.2.3 in the format of "'Part Name'.*". *1
6.	MESSAGE_TYPE	Write "OTHER".
7.	REQUEST_TYPE	Write prohibited.
8.	REQUEST_DATA	Write the three items of the parameter starting address (hexadecimal), target parameter size (hexadecimal), and "$(*.VALUE)" using the <> convention.
9.	REQUEST_DATATYPE	<DWORD><WORD><$(*.DATATYPE)>
10.	RESPONSE_TYPE	Write prohibited.
11.	RESPONSE_DATA	Write prohibited.
12.	RESPONSE_DATATYPE	Write prohibited.
13.	ERR_TYPE	Write prohibited.
14.	ERR_CODE_RANGE	Write prohibited.
15.	RELATED_MESSAGE	Write prohibited.
16.	COMMENT	Not specified. Write a value in the same manner as the regular MESSAGE part.

*1: All elements of COMM_IF_PARAMETER (all elements specified as TARGET of MESSAGE for the process in which TARGET is written with PARAMETER) that are the parameter writing targets need to be written so that they become the targets of parameter automatic setting. However, elements whose DATATYPE is constant (CONST) are out of this restriction. Therefore, some of the elements can be written only in either part.

(a) Item description example of element

```
<p:messageMember label="MessageA">
  <p:label2><p:item>MessageA</p:item></p:label2>
  <p:name><p:item>Message A</p:item></p:name>
  <p:target><p:item>WriteParam.*</p:item></p:target>
  <p:messageType><p:item>OTHER</p:item></p:messageType>
  <p:requestData>
    <p:item>item>&lt;0x0401&gt;&lt;0x0000&gt;&lt;0x00000000&gt;&lt;$0x0006&gt;</
    p:item>
  </p:requestData>
  <p:requestDatatype>
    <p:item>&lt;WORD&gt;&lt;WORD&gt;&lt;DWORD&gt;&lt;WORD&gt;</p:item>
  </p:requestDatatype>
</p:messageMember>
```

5.5.10.2.3. Description details of elements written in COMM_IF_PARAMETER part

(1) Item description of COMM_IF_PARAMETER part element

Table 5-437 Item Description Specifications of COMM_IF_PARAMETER Part Element

No.	Item Name	Description Details
1.	LABEL	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
4.	NAME	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
5.	DATATYPE	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
11.	WRITE_ORDER	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
12.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
13.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
14.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.
15.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER part.

(a) Item description example of element

```
<p:commIfParameterMember label="WriteParam">
   <p:label2><p:item>WriteParam</p:item></p:label2>
   <p:name><p:item>Write Parameter 1</p:item></p:name>
   <p:assign><p:item>RW</p:item></p:assign>
   <p:assign><p:item>RWw9</p:item></p:assign>
   <p:ref><p:item>BlockSec.BlockParam.WriteParam</p:item></p:ref>
</p:commIfParameterMember>
```

5.5.10.3 Parts and elements indicating parameters for safety communication function

This section describes the specifications of parts and elements when writing the parameters of the safety communication function of CC-Link IE TSN.

5.5.10.3.1. Overview

When writing the parameters specified in the safety specifications, write them in accordance with the specifications shown in this section. However, LABEL2 and NAME are recommended specifications, and other strings can be written. Furthermore, the items that are not described are optional. The parameters of the safety communication function are written in the COMM_IF_PARAMETER and BLOCK_PARAMETER parts.

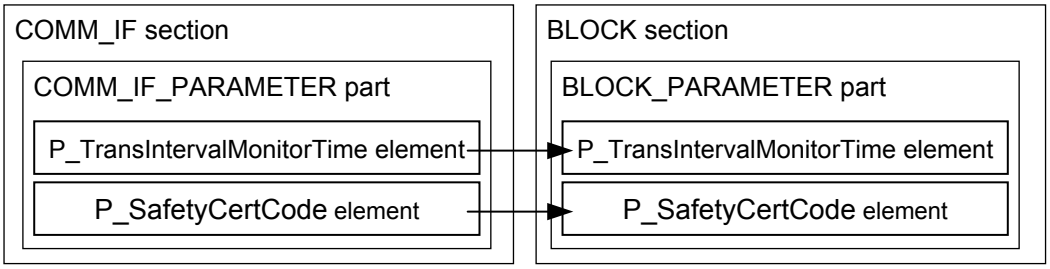


Figure 5-57 Image of Structure When Describing Safety Communication Function Parameters

Table 5-438 shows a list of elements described in the COMM_IF_PARAMETER part. For details of the description specifications of each element, see Section 5.5.10.3.2.

Table 5-438 List of Elements Written in COMM_IF_PARAMETER Part

No.	Element Name	Description Details	Required/Optional
1.	P_TransIntervalMonitorTime	Write the transmission interval monitoring time of the safety communication function.	Required
2.	P_SafetyCertCode	Write the safety certification code of the safety communication function.	Required

Table 5-439 shows a list of elements described in the BLOCK_PARAMETER part. For details of the description specifications of each element, see Section 5.5.10.3.3.

Table 5-439 List of Elements Written in BLOCK_PARAMETER Part

No.	Element Name	Description Details	Required/Optional
1.	P_TransIntervalMonitorTime	Write the transmission interval monitoring time of the safety communication function.	Required
2.	P_SafetyCertCode	Write the safety certification code of the safety communication function.	Required

5.5.10.3.2. Description details of elements written in COMM_IF_PARAMETER part

(1) Item description of P_TransIntervalMonitorTime element in COMM_IF_PARAMETER part

Table 5-440 Item Description Specifications of P_TransIntervalMonitorTime Element in COMM_IF_PARAMETER Part

No.	Item Name	Description Details
1.	LABEL	Write "P_TransIntervalMonitorTime".
2.	LABEL2	Write "P_TransIntervalMonitorTime".
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
4.	NAME	Write "Transmission Interval Monitoring Time",
5.	DATATYPE	May be omitted because this case uses the value of the reference source of the REF item. When writing, write "UINT16".
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
7.	RANGE	May be omitted because this case uses the value of the reference source of the REF item. Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	May be omitted because this case uses the value of the reference source of the REF item. When writing, write "ms".
10.	ACCESS	May be omitted because this case uses the value of the reference source of the REF item. When writing, write "W" or "RW".
11.	WRITE_ORDER	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
12.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
13.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
14.	REF	Describe the reference to the "P_TransIntervalMonitorTime" element of the BLOCK_PARAMETER part.
15.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.

(a) Item description example of element

```
<p:commIfParameterMember label="P_TransIntervalMonitorTime">
   <p:label2><p:item>P_TransIntervalMonitorTime</p:item></p:label2>
   <p:name><p:item>Transmission Interval Monitoring Time</p:item></p:name>
   <p:ref><p:item>BlockSec.BlockParameter.P_TransIntervalMonitorTime</p:item></p:
   ref>
</p:commIfParameterMember>
```

(2) Item description of P_SafetyCertCode element in COMM_IF_PARAMETER part

Table 5-441 Item Description Specifications of P_SafetyCertCode Element in COMM_IF_PARAMETER Part

No.	Item Name	Description Details
1.	LABEL	Write "P_SafetyCertCode".
2.	LABEL2	Write "P_SafetyCertCode".
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
4.	NAME	Write "Safety Certification Code".
5.	DATATYPE	May be omitted because this case uses the value of the reference source of the REF item. When writing, write "DWORD".
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
7.	RANGE	May be omitted because this case uses the value of the reference source of the REF item. Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	May be omitted because this case uses the value of the reference source of the REF item. When writing, write "RW".
11.	WRITE_ORDER	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
12.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
13.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
14.	REF	Describe the reference to the "P_SafetyCertCode" element of the BLOCK_PARAMETER part.
15.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.

(a) Item description example of element

```
<p:commIfParameterMember label="P_SafetyCertCode">
   <p:label2><p:item>P_SafetyCertCode</p:item></p:label2>
   <p:name><p:item>Safety Certification Code</p:item></p:name>
   <p:ref><p:item>BlockSec.BlockParameter.P_SafetyCertCode</p:item></p:ref>
</p:commIfParameterMember>
```

5.5.10.3.3. Description details of elements written in BLOCK_PARAMETER part

(1) Item description of P_TransIntervalMonitorTime element in BLOCK_PARAMETER part

Table 5-442 Item Description Specifications of P_TransIntervalMonitorTime Element in BLOCK_PARAMETER Part

No.	Item Name	Description Details
1.	LABEL	Write "P_TransIntervalMonitorTime".
2.	LABEL2	Write "P_TransIntervalMonitorTime".
3.	CATEGORY	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
4.	NAME	Write "Transmission Interval Monitoring Time",
5.	DATATYPE	Write "UINT16".
6.	DEFAULT	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
7.	RANGE	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write "ms".
10.	ACCESS	Write "W" or "RW".
11.	WRITE_ORDER	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
13.	COMMENT	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.

(a) Item description example of element

```
<p:blockParameterMember label="P_TransIntervalMonitorTime">
   <p:label2><p:item>P_TransIntervalMonitorTime</p:item></p:label2>
   <p:name><p:item>Transmission Interval Monitoring Time</p:item></p:name>
   <p:datatype><p:item>UINT16</p:item></p:datatype>
   <p:engUnit><p:item>ms</p:item></p:engUnit>
   <p:access><p:item>RW</p:item></p:access>
</p:blockParameterMember>
```

(2) Item description of P_SafetyCertCode element in BLOCK_PARAMETER part

Table 5-443 Item Description Specifications of P_SafetyCertCode Element in BLOCK_PARAMETER Part

No.	Item Name	Description Details
1.	LABEL	Write "P_SafetyCertCode".
2.	LABEL2	Write "P_SafetyCertCode".
3.	CATEGORY	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
4.	NAME	Write "Safety Certification Code".
5.	DATATYPE	Write "DWORD".
6.	DEFAULT	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
7.	RANGE	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
8.	MIN_INC	Write prohibited.
9.	ENG_UNIT	Write prohibited.
10.	ACCESS	Write "RW".
11.	WRITE_ORDER	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
13.	COMMENT	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.

(a) Item description example of element

```
<p:blockParameterMember label="P_SafetyCertCode">
   <p:label2><p:item>P_SafetyCertCode</p:item></p:label2>
   <p:name><p:item>Safety Certification Code</p:item></p:name>
   <p:datatype><p:item>DWORD</p:item></p:datatype>
   <p:access><p:item>RW</p:item></p:access>
</p:blockParameterMember>
```

5.5.10.4 MESSAGE specifications for writing the safety communication function parameter

5.5.10.4.1. Overview

When writing MESSAGE for writing parameters to a safety compatible device, write them in accordance with the specifications specified in this section. However, LABEL2 and NAME are recommended specifications, and other strings can be written.

Furthermore, when reading and writing the CRC32 value, write the string indicated in Table 5-444 in REQUEST_DATA and RESPONSE_DATA. The parameters of the safety communication function (see Section 5.5.10.3) are also written using the messages shown in this section.

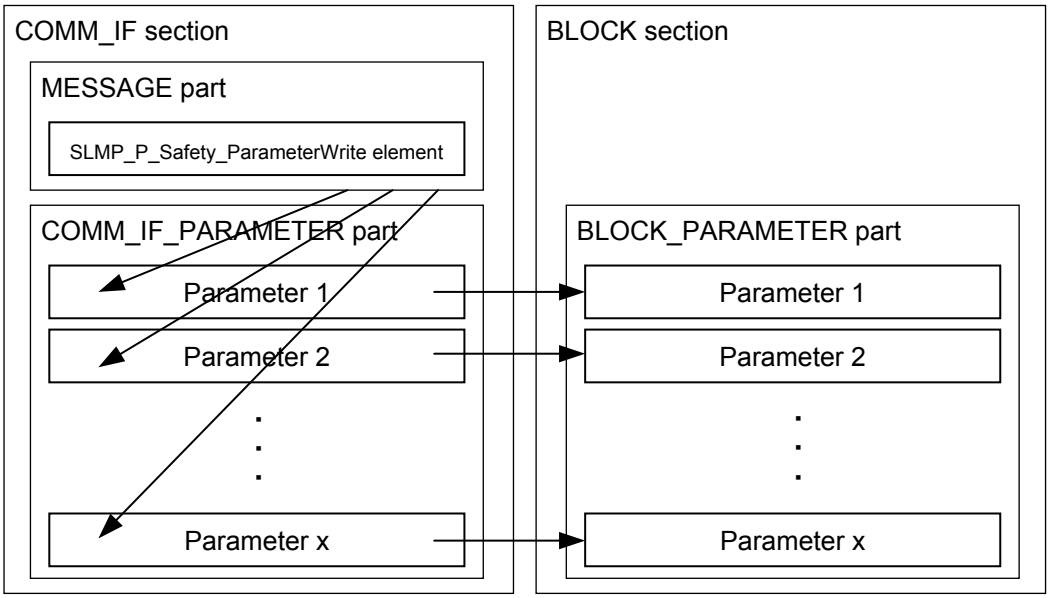


Figure 5-58 Image of Structure When Describing Parameter Writing of Safety Communication Function

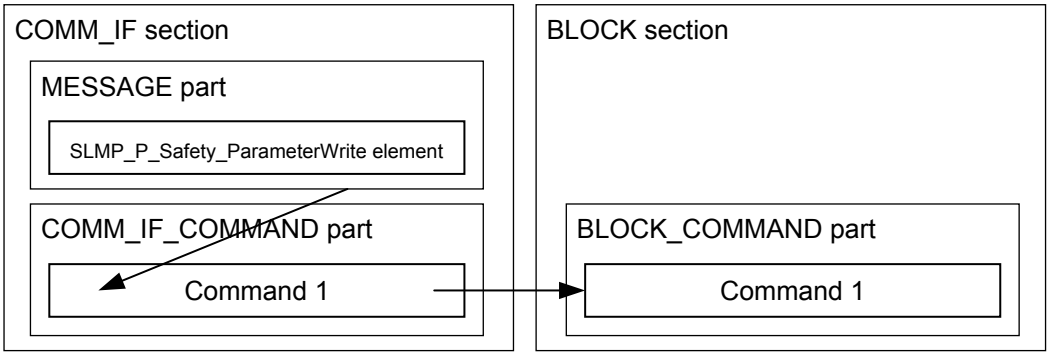


Figure 5-59 Image of Structure When Describing Enabling of Safety Communication Function Parameters

Table 5-444 Reserved Words to Indicate Reading and Writing CRC32 Value

Meaning	Reserved Word
CRC32 value specified in CC-Link IE safety communication function *1	$(CRC32_F1922815)
CRC32 value specified in IEEE802.3 *1	$(CRC32_04C11DB7)

*1: Utility software calculates CRC32 according to the following specifications.

- CRC32 is calculated from the VALUE of all elements of COMM_IF_PARAMETER (BLOCK_PARAMETER) referenced by the TARGET of the parameter writing for the safety-compliant devices (SLMP_P_Safety_ParameterWrite).
- When running multiple MESSAGE elements sequentially with RELATED_MESSAGE, CRC32 is calculated from the VALUE of all elements of COMM_IF_PARAMETER (BLOCK_PARAMETER) referenced by the TARGET of all MESSAGE elements referenced by RELATED_MESSAGE.

When reserved words indicating the value of CRC32 are written in REQUEST_DATA, utility software calculates CRC32 when MESSAGE is executed, and stores it in a request message. If it is written in RESPONSE_DATA, the utility software reads the value of CRC32 from the place where the response message reserved word is written.

When parameter writing of the safety communication function is described with multiple consolidated MESSAGEs, the element names specified in this section (SLMP_P_Safety_ParameterWrite) are attached to a MESSAGE representing a series of operations. The element names of the MESSAGE consolidated from there are optional.

A single reserved word of Table 5-444 can be written in each REQUEST_DATA or RESPONSE_DATA.

Table 5-445 shows a list of elements described in the MESSAGE part. For details of the description specifications of each element, see Section 5.5.10.4.2.

Table 5-445 List of Elements Described in MESSAGE Part

No.	Element Name	Description Details	Required/Optional
1.	SLMP_P_Safety_ParameterWrite	Write the message used for writing parameter of the safety compatible device.	Optional
2.	SLMP_P_Safety_EnableParameter	Write the parameter enabling information of the safety compatible device.	Optional

Reserved words indicating CRC32 (refer to Table 5-444) must be written to RESPONSE_DATA by the above elements or at least one element among the elements referred by RELATED_MESSAGE.

(1) Elements of parameters referenced by parameter writing of safety communication function

Table 5-446 shows a list of elements described in the COMM_IF_PARAMETER part. For details of the description specifications of each element, see Section 5.5.10.4.2.

Table 5-446 List of Elements Described in COMM_IF_PARAMETER Part

No.	Element Name	Description Details	Required/Optional
1.	(Any name)	Write the parameters of the safety communication function.	Optional

Table 5-447 shows a list of elements described in the BLOCK_PARAMETER part. For details of the description specifications of each element, see Section 5.5.10.4.2.

Table 5-447 List of Elements Described in BLOCK_PARAMETER Part

No.	Element Name	Description Details	Required/Optional
1.	(Any name)	Write the parameters of the safety communication function.	Optional

(2) Elements of commands referenced by parameter enabling of safety communication function

Table 5-448 shows a list of elements described in the COMM_IF_COMMAND part. For details of the description specifications of each element, see Section 5.5.10.4.3.

Table 5-448 List of Elements Described in COMM_IF_COMMAND Part

No.	Element Name	Description Details	Required/Optional
1.	(Any name)	Write the parameter enabling command of the safety communication function.	Optional

Table 5-449 shows a list of elements described in the BLOCK_COMMAND part. For details of the description specifications of each element, see Section 5.5.10.4.3.

Table 5-449 List of Elements Described in BLOCK_COMMAND Part

No.	Element Name	Description Details	Required/Optional
1.	(Any name)	Write the parameter enabling command of the safety communication function.	Optional

5.5.10.4.2. MESSAGE specifications for writing the safety communication function parameter

MESSAGEs are specified for parameter writing, but not for reading. MESSAGEs for reading can be written freely according to CSP+ specifications.

(1) Item description of SLMP_P_Safety_ParameterWrite element

Table 5-450 Item Description Specifications of SLMP_P_Safety_ParameterWrite Element

No.	Item Name	Description Details
1.	LABEL	Write "SLMP_P_Safety_ParameterWrite".
2.	LABEL2	Write "SLMP_P_Safety_ParameterWrite". [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular MESSAGE part.
4.	NAME	Write "Parameter Write". [Recommended specifications]
5.	TARGET	Write the reference to the COMM_IF_PARAMETER part in which the parameters of the safety communication function are written.
6.	MESSAGE_TYPE	Write "PARAMETER".
7.	REQUEST_TYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
8.	REQUEST_DATA	Not specified. Write a value in the same manner as the regular MESSAGE part.
9.	REQUEST_DATATYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
10.	RESPONSE_TYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
11.	RESPONSE_DATA	Not specified. Write a value in the same manner as the regular MESSAGE part.
12.	RESPONSE_DATATYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
13.	ERR_TYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
14.	ERR_CODE_RANGE	Not specified. Write a value in the same manner as the regular MESSAGE part.
15.	RELATED_MESSAGE	Not specified. Write a value in the same manner as the regular MESSAGE part.
16.	COMMENT	Not specified. Write a value in the same manner as the regular MESSAGE part.

(a) Item description example of SLMP_P_Safety_ParameterWrite element

```
<p:messageMember label="SLMP_P_Safety_ParameterWrite">
   <p:label2><p:item>SLMP_P_Safety_ParameterWrite</p:item></p:label2>
   <p:name><p:item>Parameter Write</p:item></p:name>
   <p:target><p:item>SEQ TARGET</p:item></p:target>
   <p:messageType><p:item>PARAMETER</p:item></p:messageType>
   <p:relatedMessage>
      <p:item>&lt;SEQ SLMPStartWritePramArea&gt;&lt;SEQ
      SLMPStationWritePrm&gt;&lt;SEQ SLMPWriteCRC32&gt;&lt;SEQ
      SLMPReflectPrm&gt;&lt;SEQ SLMPStartReadPramArea&gt;&lt;SEQ
      SLMPReadCRC32&gt;</p:item>
   </p:relatedMessage>
</p:messageMember>
```

(2) Item description of COMM_IF_PARAMETER part element

Table 5-451 Item Description Specifications of COMM_IF_PARAMETER Part Element

No.	Item Name	Description Details
1.	LABEL	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
4.	NAME	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
5.	DATATYPE	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
7.	RANGE	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
10.	ACCESS	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
11.	WRITE_ORDER	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
12.	ASSIGN	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
13.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
14.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.
15.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_PARAMETER.

(a) Item description example of element

```
<p:commIfParameterMember label="Parameter1">
   <p:label2><p:item>Parameter1</p:item></p:label2>
   <p:name><p:item>Parameter 1</p:item></p:name>
   <p:access><p:item>RW</p:item></p:access>
   <p:assign><p:item>RWr1</p:item></p:name>
   <p:ref><p:item>BlockSec.BlockParameter.Parameter1</p:item></p:ref>
</p:commIfParameterMember>
```

(3) Item description of BLOCK_PARAMETER part element

Table 5-452 Item Description Specifications of BLOCK_PARAMETER Part Element

No.	Item Name	Description Details
1.	LABEL	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
2.	LABEL2	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
4.	NAME	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
5.	DATATYPE	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
6.	DEFAULT	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
7.	RANGE	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
8.	MIN_INC	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
9.	ENG_UNIT	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
10.	ACCESS	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
11.	WRITE_ORDER	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
12.	UI_ATTRIBUTE	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.
13.	COMMENT	Not specified. Write a value in the same manner as the regular BLOCK_PARAMETER.

(a) Item description example of element

```
<p:blockParameterMember label="Parameter1">
   <p:label2><p:item>Parameter1</p:item></p:label2>
   <p:name><p:item>Parameter 1</p:item></p:name>
   <p:datatype><p:item>INT16</p:item></p:datatype>
   <p:range><p:item>[0,200]</p:item></p:range>
   <p:engUnit><p:item>ms</p:item></p:engUnit>
   <p:access><p:item>RW</p:item></p:access>
</p:blockParameterMember>
```

5.5.10.4.3. MESSAGE specifications for parameter enabling of safety communication function

(1) Item description of SLMP_P_Safety_EnableParameter element

Table 5-453 Item Description Specifications of SLMP_P_Safety_EnableParameter Element

No.	Item Name	Description Details
1.	LABEL	Write "SLMP_P_Safety_EnableParameter".
2.	LABEL2	Write "SLMP_P_Safety_EnableParameter". [Recommended specifications]
3.	CATEGORY	Not specified. Write a value in the same manner as the regular MESSAGE part.
4.	NAME	Write "Enable Parameter". [Recommended specifications]
5.	TARGET	Write the reference to the COMM_IF_COMMAND part in which the commands of the safety communication function are written.
6.	MESSAGE_TYPE	Write "COMMAND".
7.	REQUEST_TYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
8.	REQUEST_DATA	Not specified. Write a value in the same manner as the regular MESSAGE part.
9.	REQUEST_DATATYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
10.	RESPONSE_TYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
11.	RESPONSE_DATA	Not specified. Write a value in the same manner as the regular MESSAGE part.
12.	RESPONSE_DATATYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
13.	ERR_TYPE	Not specified. Write a value in the same manner as the regular MESSAGE part.
14.	ERR_CODE_RANGE	Not specified. Write a value in the same manner as the regular MESSAGE part.
15.	RELATED_MESSAGE	Not specified. Write a value in the same manner as the regular MESSAGE part.
16.	COMMENT	Not specified. Write a value in the same manner as the regular MESSAGE part.

(a) Item description example of SLMP_P_Safety_EnableParameter element

```
<p:messageMember label="SLMP_P_Safety_EnableParameter">
        <p:label2>
                <p:item>EnableParameter</p:item>
        </p:label2>
        <p:name>
                <p:item>Enable Parameter</p:item>
        </p:name>
        <p:target>
                <p:item>SEQ TARGET</p:item>
        </p:target>
        <p:messageType>
                <p:item>COMMAND</p:item>
        </p:messageType>
        <p:relatedMessage>
                <p:item>&lt;SEQ SLMPWriteCRC32&gt;&lt;SEQ
SLMPCheckCRC32&gt;&lt;SEQ SLMPReadCRC32&gt;&lt;SEQ
SLMPSafetyEnableParameter&gt;</p:item>
        </p:relatedMessage>
</p:messageMember>
```

(2) Item description of COMM_IF_COMMAND part element

Table 5-454 Item Description Specifications of COMM_IF_COMMAND Part Element

No.	Item Name	Description Details
1.	LABEL	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.
2.	LABEL2	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.
4.	NAME	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.
5.	ARGUMENT	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.
6.	REF	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.
7.	COMMENT	Not specified. Write a value in the same manner as the regular COMM_IF_COMMAND part.

(a) Item description example of P_GetStatusCommand element

```
<p:commIfCommandMember label="Safety_EnableCommand">
   <p:label2><p:item>Safety_EnableCommand</p:item></p:label2>
   <p:name><p:item>Safety Communication Command</p:item></p:name>
   <p:ref><p:item>BlockSec.BlockCommand.Safety_EnableCommand</p:item></p:ref>
</p:commIfCommandMember>
```

(3) Item description of BLOCK_COMMAND part element

Table 5-455 Item Description Specifications of BLOCK_COMMAND Part Element

No.	Item Name	Description Details
1.	LABEL	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.
2.	LABEL2	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.
3.	CATEGORY	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.
4.	NAME	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.
5.	ARGUMENT	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.
6.	COMMENT	Not specified. Write a value in the same manner as the regular BLOCK_COMMAND part.

(a) Item description example of element

```
<p:blockCommandMember label="Safety_EnableCommand">
   <p:label2><p:item>Safety_EnableCommand</p:item></p:label2>
   <p:name><p:item>Safety Communication Command</p:item></p:name>
</p:blockCommandMember>
```

6. CSP+ Description Method

6.1 CSP+ Version Management Method

Figure 6-1 shows an example of updating the profile description for one device. The following describes the rules for updating the profile description and managing the updated file.

(1) When you create a CSP+ file in which the device version has been updated, manage the file as indicated below in accordance with the value of the VersionPolicyType element [see Section 5.2.1.3(9)].

[When the VersionPolicyType element is 0 (When only the latest CSP+ file is managed)]

Update and manage only the CSP+ file of the latest device version [Recommended specifications].

[When the VersionPolicyType element is a value other than 0 (When a CSP+ file is managed for each device version)]

Manage the CSP+ file of the old device version and the CSP+ file of the new device version in parallel [Required]. Then, update the CSP+ file of the old device version as required [Recommended specifications].

* The utility software uses the mass production device in accordance with the device version described in the CSP+ file. As a result, the old CSP+ file needs to be managed as well.

(2) When you create a CSP+ file in which the minor version of the CSP+ specification version is updated, manage only the CSP+ file of the new version.

* For details on the major version and minor version of the CSP+ specification version, see Section 5.1.1.3(6).

(3) When you create a CSP+ file in which the major version of the CSP+ specification version is updated, you need to manage the old CSP+ file as well.

* Note, however, that in a case where the file does not need to be used with utility software that does not support the new major version, there is no need to manage the old CSP+ file.

(4) CSP+ files for the same device version need to be managed so that there are no duplicate file versions [Required].

(5) For CSP+ files in which only the language-dependent information differs while all other information is the same, make the major numbers of the file versions the same [Required].

* For details on the major version and minor version of the file version, see Section 5.1.1.3(7).

* Utility software that supports other languages determines the target CSP+ file by the device version and major version of the file version only.

* For example, in a case where only a Japanese portion of a Japanese version CSP+ file needs to be corrected and the CSP+ files of other languages do not, increase the minor number of the file version of the Japanese version CSP+ file and leave the major version as is.

(6) For CSP+ files in which the device versions are the same and the major versions of the CSP+ specification versions are the same, the CSP+ file with the higher file version is regarded as newer and used with priority by the utility software [Required].

* The priority order between CSP+ files having different major versions is determined by the file version alone.

(7) When a device version is increased, set the the file version as 1. [Recommended specifications].

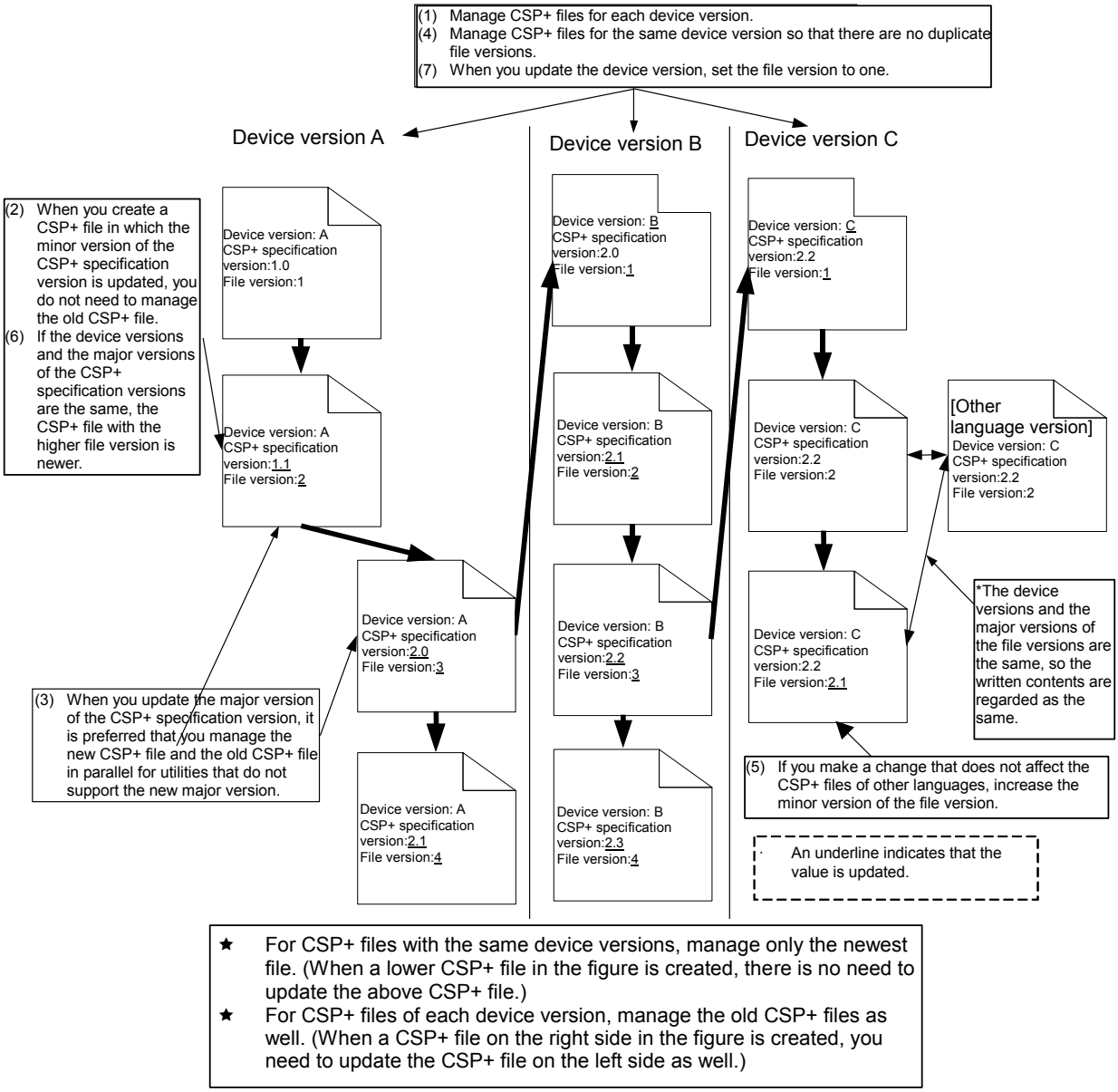


Figure 6-1 Version Update Example

6.2 Other Language Management

When you want to support a CSP+ file in another language for a device of the same model, change only the items below. Do not change any other item. [Recommended specifications]

- LABEL2
- NAME
- CATEGORY
- DATA
- COMMENT
- ENG_UNIT

* While substantially all units do not need to be changed, change is required for those units written using double-byte characters, not alphanumeric characters, such as "Ω".

In addition, as indicated in this section, for CSP+ files that simply have different language-dependent information with all other information being the same, write the same major version [see Section 5.1.1.3(7)] in the file version for the CSP+ files.

6.3 CSP+ File Management

The utility software uses the following seven types of information when identifying a CSP+ file:

[1] Model

[2] Vendor code

[3] Supported language information

[4] Device version

[5] CSP+ specification version

[6] File version

[7] Device configuration ID

Table 6-1 describes each identifier as well as the methods used by the utility software for managing CSP+ files having different identifiers. The CSP+ file creator and manager need to create a profile description so that the utility software can execute management as shown in Table 6-1. In particular, caution is required during device version management since the CSP+ files of older version also require management.

Table 6-1 Information for Identifying CSP+ File and Management Method by Utility Software

No.	Type of Identifier	Management Method by Utility Software
1.	Model (See DeviceModel of DEVICE_INFO in Section 5.2.1.3.)	Manage the CSP+ files of all models separately (recommended).
2.	Vendor code (See VendorCode of DEVICE_INFO in Section 5.2.1.3.)	Manage the CSP+ files of all vendor codes separately (recommended).
3.	Supported language information (See Language of FILE_INFO of Section 5.1.1.3.)	Manage the CSP+ files of all language information separately (recommended).
4.	Device version (See Version of DEVICE_INFO of Section 5.2.1.3.)	Manage the CSP+ files of all device versions separately (recommended). However, in a case of a device in which a CSP+ file of the latest device version can be used [a device in which the VersionPolicyType (See Section 5.2.1.3(9) is 0), manage only the profile of the newest device version.
5.	CSP+ specification version (See CCLinkFamilyProfileVersion of FILE_INFO of Section 5.1.1.3.)	Manage the CSP+ file having the highest version among all of the versions of CSP+ specifications supported by the utility software (recommended).
6.	File version (See FileVersion of FILE_INFO of Section 5.1.1.3.)	Manage only the newest CSP+ file (recommended).
7.	Device configuration ID (See DeviceConfigurationID of DEVICE_INFO of Section 5.2.1.3.)	Manage CSP+ files with/without information and with content differences separately (recommended).

7. Control & Communication System Profile XML Scheme

7.1 XML Schema

The following describes the XML schema for CSP+.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
               elementFormDefault="qualified"
               targetNamespace="http://cc-link.org/cspplus/ver3/"
               xmlns:p="http://cc-link.org/cspplus/ver3/">

       <!--*********************************************-->
       <!--[Overall Structure]-->
       <!--*********************************************-->


     <?xml version="1.0" encoding="utf-8"?>
               <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
               elementFormDefault="qualified"
               targetNamespace="http://cc-link.org/cspplus/ver2/"
               xmlns:p="http://cc-link.org/cspplus/ver2/">
       <!--*********************************************-->
       <!—[Overall Structure] -->
       <!--*********************************************-->

       <xsd:element name="profile">
         <xsd:complexType>
           <xsd:sequence>
             <xsd:element ref="p:file" minOccurs="1" maxOccurs="1"/>
             <xsd:element ref="p:device" minOccurs="1" maxOccurs="1"/>
             <xsd:element ref="p:block" minOccurs="0" maxOccurs="unbounded"/>
             <xsd:element ref="p:commIf" minOccurs="0" maxOccurs="unbounded"/>
           </xsd:sequence>
         </xsd:complexType>
       </xsd:element>

       <!--*********************************************-->
       <!—[Structure of each section] -->
       <!--*********************************************-->

       <!-- Sections -->
       <xsd:element name="file">
         <xsd:complexType>
           <xsd:sequence>
             <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
             <xsd:element ref="p:fileInfo" minOccurs="1" maxOccurs="1"/>
           </xsd:sequence>
           <xsd:attribute name="label" type="p:LabelType" use="required"/>
         </xsd:complexType>
       </xsd:element>

       <xsd:element name="device">
         <xsd:complexType>
           <xsd:sequence>
             <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
             <xsd:element ref="p:deviceInfo" minOccurs="1" maxOccurs="1"/>
           </xsd:sequence>
           <xsd:attribute name="label" type="p:LabelType" use="required"/>
         </xsd:complexType>
       </xsd:element>

       <xsd:element name="block">
         <xsd:complexType>
           <xsd:sequence>
             <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
             <xsd:element ref="p:blockInfo" minOccurs="1" maxOccurs="1"/>
             <xsd:element ref="p:blockInput" minOccurs="0" maxOccurs="unbounded"/>
             <xsd:element ref="p:blockOutput" minOccurs="0" maxOccurs="unbounded"/>
             <xsd:element ref="p:blockParameter" minOccurs="0" maxOccurs="unbounded"/>
```

```
      <xsd:element ref="p:blockCommand" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="p:struct" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="p:enum" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="p:commandArgument" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="p:condition" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="commIf">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:commIfInfo" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="p:commIfInput" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="p:commIfOutput" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="p:commIfParameter" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="p:commIfCommand" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="p:method" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="p:message" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="p:struct" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="p:enum" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="p:commandArgument" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<!--*******************************************-->
<!—[Structure of various parts] -->
<!--*******************************************-->

<!—Defines the Member structure of parts for INFO-->
<xsd:complexType name="InfoTableMemberType">
  <xsd:sequence>
    <xsd:element ref="p:label2" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:category" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:name" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:datatype" minOccurs="1" maxOccurs="1"/>
    <xsd:element ref="p:data" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="label" type="p:LabelType" use="required"/>
</xsd:complexType>

<!—Defines structure of FILE_INFO part-->
<xsd:element name="fileInfo">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:fileInfoMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="fileInfoMember" type="p:InfoTableMemberType"/>

<!--Defines structure of DEVICE_INFO part-->
<xsd:element name="deviceInfo">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:deviceInfoMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="deviceInfoMember" type="p:InfoTableMemberType"/>

<!--Defines structure of BLOCK_INFO part-->
<xsd:element name="blockInfo">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:blockInfoMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="blockInfoMember" type="p:InfoTableMemberType"/>

<!--Defines structure of COMM_IF_INFO part-->
<xsd:element name="commIfInfo">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:commIfInfoMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="commIfInfoMember" type="p:InfoTableMemberType"/>

<!-- Defines Member structure of BLOCK_INPUT, BLOCK_OUTPUT parts -->
<xsd:complexType name="BlockTableMemberType">
  <xsd:sequence>
    <xsd:element ref="p:label2" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:category" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:name" minOccurs="1" maxOccurs="1"/>
    <xsd:element ref="p:datatype" minOccurs="1" maxOccurs="1"/>
    <xsd:element ref="p:default" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:range" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:minInc" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:engUnit" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:access" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:uiAttribute" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="label" type="p:LabelType" use="required"/>
</xsd:complexType>

<!—Defines structure of BLOCK_INPUT parts-->
<xsd:element name="blockInput">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:blockInputMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="blockInputMember" type="p:BlockTableMemberType"/>

<!—Defines structure of BLOCK_OUTPUT parts-->
<xsd:element name="blockOutput">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:blockOutputMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="blockOutputMember" type="p:BlockTableMemberType"/>

<!-- Defines structure of BLOCK_PARAMETER part-->
<xsd:element name="blockParameter">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:blockParameterMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="blockParameterMember">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:label2" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:category" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:name" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="p:datatype" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="p:default" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:range" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:minInc" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:engUnit" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:access" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="p:writeOrder" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:uiAttribute" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<!--Defines structure of BLOCK_COMMAND part-->
<xsd:element name="blockCommand">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:blockCommandMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="blockCommandMember">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:label2" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:category" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:name" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="p:argument" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<!-- Defines structure of COMM_IF_INPUT, COMM_IF_OUTPUT parts -->
<xsd:complexType name="CommIfTableMemberType">
  <xsd:sequence>
    <xsd:element ref="p:label2" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:category" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:name" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:datatype" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:default" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:range" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:minInc" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:engUnit" minOccurs="0" maxOccurs="1"/>
```

```
    <xsd:element ref="p:access" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:assign" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:uiAttribute" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:ref" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="label" type="p:LabelType" use="required"/>
</xsd:complexType>

<!-- Defines structure of COMM_IF_INPUT part -->
<xsd:element name="commIfInput">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:commIfInputMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="commIfInputMember" type="p:CommIfTableMemberType"/>

<!-- Describes structure of COMM_IF_OUTPUT part -->
<xsd:element name="commIfOutput">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:commIfOutputMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="commIfOutputMember" type="p:CommIfTableMemberType"/>

<!-- Defines structure of COMM_IF_PARAMETER part -->
<xsd:element name="commIfParameter">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:commIfParameterMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="commIfParameterMember">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:label2" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:category" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:name" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:datatype" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:default" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:range" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:minInc" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:engUnit" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:access" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:writeOrder" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:assign" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:uiAttribute" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:ref" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<!—Defines structure of COMM_IF_COMMAND part -->
```

```
<xsd:element name="commIfCommand">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:commIfCommandMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="commIfCommandMember">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:label2" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:category" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:name" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:argument" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:ref" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<!—Defines structure of METHOD part-->
<xsd:element name="method">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:methodMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="methodMember">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:label2" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:category" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:name" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="p:target" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="p:methodType" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="p:writeRegister" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:writeData" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:writeDatatype" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:readRegister" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:readData" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:readDatatype" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:interlock" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:reqFlag" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:endCondition" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:errCondition" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:errRegister" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:errCodeRange" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:relatedMethod" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<!—Defines structure of MESSAGE part-->
<xsd:element name="message">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:messageMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
```

```
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="messageMember">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:label2" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:category" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:name" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="p:target" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="p:messageType" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="p:requestType" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:requestData" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:requestDatatype" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:responseType" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:responseData" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:responseDatatype" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:errType" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:errCodeRange" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:relatedMessage" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<!-- Common information -->
<!—Defines structure of STRUCT part-->
<xsd:element name="struct">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:structMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="structMember">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:label2" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:category" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:name" minOccurs="0" maxOccurs="1"/>
       <xsd:element ref="p:datatype" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:default" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:range" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:minInc" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:engUnit" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:offset" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="p:ref" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<!—Defines structure of ENUM part —>
<xsd:element name="enum">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:enumMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="enumMember">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:label2" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:category" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:name" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="p:code" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="p:relatedEle" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<!—Defines structure of COMMAND_ARGUMENT part-->
<xsd:element name="commandArgument">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:commandArgumentMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="commandArgumentMember">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:label2" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:category" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:name" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:datatype" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:default" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:range" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:minInc" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:engUnit" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:access" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="p:assign" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:ref" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="p:LabelType" use="required"/>
  </xsd:complexType>
</xsd:element>

<!—Defines structure of CONDITION part-->
<xsd:element name="condition">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:comment" minOccurs="0" maxOccurs="1"/>
      <!--Describes reference information to labels used for expressions-->
      <xsd:element ref="p:conditionVariable" minOccurs="1" maxOccurs="1"/>
      <!--Writes values used for conditional judgment-->
      <xsd:element ref="p:conditionMember" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="label" type="xsd:string" use="required"/>
    <xsd:attribute name="conditionNumber" type="xsd:positiveInteger" use="required"/>
    <xsd:attribute name="resultNumber" type="xsd:positiveInteger" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="conditionVariable">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:conditionXVariable" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="conditionMember">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="p:conditionX" minOccurs="1" maxOccurs="unbounded"/>
      <xsd:element ref="p:resultX" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!--*********************************************-->
<!—[Item type declaration] -->
<!--*********************************************-->
<xsd:element name="label2" type="p:ItemLabel2Type"/>
<xsd:element name="category" type="p:ItemType"/>
<xsd:element name="name" type="p:ItemType"/>
<xsd:element name="datatype" type="p:ItemOrStructType"/>
<xsd:element name="default" type="p:ItemType"/>
<xsd:element name="range" type="p:ItemListOrEnumOrConditionType"/>
<xsd:element name="minInc" type="p:ItemOrConditionType"/>
 <xsd:element name="engUnit" type="p:ItemOrConditionType"/>
<xsd:element name="assign" type="p:ItemType"/>

<xsd:element name="access" type="p:ItemOrConditionType"/>
<xsd:element name="ref" type="p:ItemFLabelListType"/>
<xsd:element name="comment" type="p:IndexedItemType"/>
<xsd:element name="uiAttribute" type="p:ItemListType"/>
<xsd:element name="writeOrder" type="p:ItemType"/>
<xsd:element name="data" type="p:ItemListType"/>
<xsd:element name="argument" type="p:ItemType"/>
<!—Items for Method -->
<xsd:element name="target" type="p:ItemListType"/>
<xsd:element name="methodType" type="p:ItemListType"/>
<xsd:element name="writeRegister" type="p:ItemType"/>
<xsd:element name="writeData" type="p:ItemType"/>
<xsd:element name="writeDatatype" type="p:ItemType"/>
<xsd:element name="readRegister" type="p:ItemType"/>
<xsd:element name="readData" type="p:ItemType"/>
<xsd:element name="readDatatype" type="p:ItemType"/>
<xsd:element name="interlock" type="p:ItemType"/>
<xsd:element name="reqFlag" type="p:ItemListType"/>
<xsd:element name="endCondition" type="p:ItemListType"/>
<xsd:element name="errCondition" type="p:ItemType"/>
<xsd:element name="errRegister" type="p:ItemType"/>
<xsd:element name="errCodeRange" type="p:ItemListOrEnumType"/>
<xsd:element name="relatedMethod" type="p:ItemListType"/>
<!—Items for Message-->
<xsd:element name="messageType" type="p:ItemListType"/>
<xsd:element name="requestType" type="p:ItemType"/>
<xsd:element name="requestData" type="p:ItemType"/>
<xsd:element name="requestDatatype" type="p:ItemType"/>
<xsd:element name="responseType" type="p:ItemType"/>
<xsd:element name="responseData" type="p:ItemType"/>
<xsd:element name="responseDatatype" type="p:ItemType"/>
<xsd:element name="errType" type="p:ItemType"/>
<xsd:element name="relatedMessage" type="p:ItemListType"/>
<!—Items for common information-->
<xsd:element name="offset" type="p:ItemType"/>
<xsd:element name="code" type="p:ItemType"/>
<xsd:element name="relatedEle" type="p:ItemType"/>
<xsd:element name="conditionXVariable" type="p:IndexedItemType"/>
<xsd:element name="conditionX" type="p:IndexedItemListType"/>
<xsd:element name="resultX" type="p:IndexedItemListOrEnumType"/>

<!--*********************************************-->
<!—[Type Definition] -->
<!--*********************************************-->

<!--Type for items that only describe “item”-->
<xsd:complexType name="ItemType">
```

```
  <xsd:sequence>
    <xsd:element ref="p:item" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<!--Type for items that can describe multiple “items”-->
<xsd:complexType name="ItemListType">
  <xsd:sequence>
    <xsd:element ref="p:item" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<!--Type definition for items that can write references to “item” or STRUCT-->
<xsd:complexType name="ItemOrStructType">
  <xsd:choice>
    <xsd:element ref="p:item"/>
    <xsd:element ref="p:structRefItem"/>
  </xsd:choice>
</xsd:complexType>

<!--Type definition for items that can write references to “item” or CONDITION-->
<xsd:complexType name="ItemOrConditionType">
  <xsd:choice>
    <xsd:element ref="p:item"/>
    <xsd:element ref="p:conditionRefItem"/>
  </xsd:choice>
</xsd:complexType>

<!--Item definition that can write references to multiple description of “item” or
ENUM or references to CONDITION-->
<xsd:complexType name="ItemListOrEnumType">
  <xsd:choice>
    <xsd:element ref="p:item" minOccurs="1" maxOccurs="unbounded"/>
    <xsd:element ref="p:enumRefItem"/>
  </xsd:choice>
</xsd:complexType>

<!--Item definition that can write multiple description of “item” or references to
ENUM or references to CONDITION-->
<xsd:complexType name="ItemListOrEnumOrConditionType">
  <xsd:choice>
    <xsd:element ref="p:item" minOccurs="1" maxOccurs="unbounded"/>
    <xsd:element ref="p:enumRefItem"/>
    <xsd:element ref="p:conditionRefItem"/>
  </xsd:choice>
</xsd:complexType>

<!--Type for items that describe only “item” for LABEL-->
<xsd:complexType name="ItemLabel2Type">
  <xsd:sequence>
    <xsd:element name="item" type="p:Label2Type" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<!--Type that can describe multiple reference “items” with LABEL-->
<xsd:complexType name="ItemFLabelListType">
  <xsd:sequence>
    <xsd:element name="item" type="p:FLabelType" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<!--Type for items that have “index” and can only use “item”—>
<xsd:complexType name="IndexedItemType">
  <xsd:sequence>
    <xsd:element ref="p:item" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="index" type="xsd:positiveInteger" default="1" />
</xsd:complexType>
```

```
  <!--Type for items that have “index” and define multiple “items”-->
  <xsd:complexType name="IndexedItemListType">
    <xsd:sequence>
      <xsd:element ref="p:item" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="index" type="xsd:positiveInteger" default="1" />
  </xsd:complexType>

  <!--Type for items that have “index” and can describe multiple “items” or can write
  reference to Enum-->
  sd:complexType name="IndexedItemListOrEnumType">
    <xsd:choice>
      <xsd:element ref="p:item" minOccurs="1" maxOccurs="unbounded"/>
      <xsd:element ref="p:enumRefItem"/>
    </xsd:choice>
    <xsd:attribute name="index" type="xsd:positiveInteger" default="1" />
  </xsd:complexType>

  <xsd:element name="item">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:minLength value="1"/>
      </xsd:restriction>
     </xsd:simpleType>
  </xsd:element>
  <xsd:element name="structRefItem" type="p:FLabelType"/>
  <xsd:element name="enumRefItem" type="p:FLabelType"/>
  <xsd:element name="conditionRefItem" type="p:FLabelType"/>

  <xsd:simpleType name="LabelType">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[a-zA-Z][a-zA-Z0-9_]{0,31}"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="FLabelType">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[a-zA-Z][a-zA-Z0-9_]*(¥.[a-zA-Z][a-zA-Z0-9_]*)*"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="Label2Type">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[^0-9_!&quot;#$%&amp;'()*+,¥-¥./:;&lt;=&gt;?@¥[¥¥¥]^`{|}~
][^!&quot;#$%&amp;'()*+,¥-¥./:;&lt;=&gt;?@¥[¥¥¥]^`{|}~ ]{0,31}"/>
    </xsd:restriction>
  </xsd:simpleType>

</xsd:schema>
```

8. Description Specifications of Object Dictionary File

8.1 CSV File Format

Table 8- shows the format of the CSV file to write the object dictionary.

Table 8-1 CSV File Format

No.	Line No.	Details	Description
1.	1st line	Header part	Write the item name of each column. Each of the item names are shown in the Item Name column of Table 8-2 Item List.
2.	2nd line or later	Data part	Write one entry information per line. Write the items in the column order that they are shown in the header part.

The following shows the precautions on creating a CSV file.

- Do not information other than that shown in Table 8- in the CSV file.
- When changing the CSV file, change the descriptions of the elements for the device version and firmware version of the CSP+ file as necessary.
- Use Unicode (with UTF-16 little endian BOM) for the file format.
- Use a comma (,) for the delimiter between items.
- If a comma (,) or a double quotation mark (") is included in an item (which can be used only for a comment in Table 8-2 Item List), the item is enclosed double quotation marks and the double quotation mark (") in the item is represented by two double quotation marks ("").
- Set a line feed (line feed code: CR+LF) at the end of a line.

Figure 8-1 CSV File Output Example shows an output example of a CSV file. The descriptions of some objects (entries) are omitted in this example.

```
Index,SubIndex,ObjectCode,DataType,BitLength,Access,Mapping,UnitType,DefaultValue,MinValue,MaxValue,Name,LabelName,Comment
0x1600,0x00,0x09,0x0005,0x08,RW,0x0000,,0x04,0x00,0x20,1st Receive PDO Mapping,,
0x1600,0x01,0x09,0x0007,0x20,RW,0x0000,,0x20000120,,,Mapping d Object 001,,
0x1600,0x02,0x09,0x0007,0x20,RW,0x0000,,0x20000120,,,Mapping d Object 002,,
0x1A00,0x00,0x09,0x0005,0x08,RW,0x0000,,0x04,0x00,0x20,1st Transmit PDO Mapping,,
0x1A00,0x01,0x09,0x0007,0x20,RW,0x0000,,0x20000120,,,Mapping d Object 001,,
0x1A00,0x02,0x09,0x0007,0x20,RW,0x0000,,0x20000120,,,Mapping d Object 002,,
```

Figure 8-1 CSV File Output Example

8.2 Description Specifications for Each Item

Table 8-2 Item List lists the items of objects written in the CSV file.

Table 8-2 Item List

No.	Item Name	Description Details	Required/ Optional
1.	Index	Write the address (index) for accessing the object in a 2-byte numerical value (hexadecimal).	Required
2.	SubIndex	Write the address (subindex) for accessing the entry in a 1-byte numerical value (hexadecimal).	Required
3.	ObjectCode	Write the structure (object code) of the entry in a 1-byte numerical value (hexadecimal). 0x05: DEFTYPE 0x07: VAR 0x08: ARRAY 0x09: RECORD	Required
4.	DataType	Write the data type of the entry in a 2-byte numerical value (hexadecimal). 0x0002: INTEGER8 0x0003: INTEGER16 0x0004: INTEGER32 0x0005: UNSIGNED8 0x0006: UNSIGNED16 0x0007: UNSIGNED32 0x0009: VISIBLE_STRING	Required
5.	BitLength	Write the bit length of the entry in a 2-byte numerical value (hexadecimal).	Required
6.	Access	Write the access attribute of the entry in a string. R: Only reading is possible. W: Only writing is possible. RW: Both reading and writing are possible. NA: Access is not possible.	Required
7.	Mapping	Write the mapping availability in a 2-byte numerical value (hexadecimal). Specify the mapping availability with the following bits. Bit 0 to 5: Reserved (fixed to 0) Bit 6: RPDO mapping availability (0: Unavailable, 1: Available) Bit 7: TPDO mapping availability (0: Unavailable, 1: Available) Bit 8 to 15: Reserved (fixed to 0)	Required
8.	UnitType	Write the entry unit in a 4-byte numerical value (hexadecimal). For details on how to represent units, see "CiA 303-2:2012, CANopen recommendation - Part 2: Representation of SI units and prefixes".	Optional
9.	DefaultValue	Write the default value of the entry. The value to be written as the default value needs to follow DataType set for the entry. However, if it follows DataType, a value that is outside the range set by MinValue and MaxValue can be written.	Optional
10.	MinValue	Write the minimum value of the entry. The value to be written as the minimum value needs to follow DataType set for the entry. If the minimum value to be set is equal to the minimum value determined by DataType, this can be omitted.	Optional
11.	MaxValue	Write the maximum value of the entry. The value to be written as the maximum value needs to follow DataType set for the entry. If the maximum value to be set is equal to the maximum value determined by DataType, this can be omitted.	Optional

No.	Item Name	Description Details	Required/ Optional
12.	Name	Write the name of the entry. Write the name of the object when the value of SubIndex is "0x00" and write the name of the entry using an arbitrary string when the value of SubIndex is other than "0x00". Use of a comma (,) and double-quotation mark (") is prohibited.	Required
13.	LabelName	Write a string using up to 32 characters for the label name of the entry. The characters that are not permitted are shown in "Characters Not Permitted in LABEL Item" of Table 4-13. Write a entry label name so that it does not duplicate another one.	Optional
14.	Comment	Write entry descriptions and precautions on use using an arbitrary string.	Optional

9. Additional specifications for CSP+ Specifications Version 1.1 or later

Conventions added in version 1.1 or later are shown here.

If using the following description, then CSP+ specification version (See Section 5.1.1.3 (6)) must be described in a version that supports those descriptions.

9.1 Additional specifications for CSP+ Specifications Version 1.1

If CSP+ specification version is written in 1.0, the following conventions cannot be described.

(1) Consecutive execution of multiple methods or messages using reserved word SEQ
Consecutive execution of multiple methods or messages using reserved word SEQ cannot be described in version 1.0. Specifications relating to this convention are mainly written in the following places.
4.3.1.32. RELATED_MESSAGE conventions
4.3.1.33. RELATED_METHOD conventions
4.3.1.42. TARGET conventions

Consecutive execution of multiple methods or messages using reserved word PRE can be described in version 1.0.

9.2 Additional specifications for CSP+ Specifications Version 2.0

If CSP+ specification version is written in 1.1 or earlier, description of the following conventions and specifications cannot be performed.

(1) Conventions of expression by CONDITION parts
Conventions using CONDITION parts to change values of parameter items (RANGE, MIN_INC, ENG_UNIT, ACCESS) according to values of other parameter cannot be used in version 1.1 or earlier. Specifications relating to this convention are mainly written in the following places.
4.3.1.7. CONDITIONx conventions
4.3.1.8. CONDITIONXVARIABLE conventions
0. RESULTx conventions
5.4.9. CONDITION parts

(2) Conventions for max value and min value of RANGE items by reference to values
Conventions that reference values of other parameters as max value or min value of RANGE items cannot be described in version 1.1 or earlier. Specifications relating to this convention are mainly written in the following places.
4.3.1.26 RANGE conventions (8)

(3) Description specifications for the safety communication function support unit
Conventions relating to the safety communication function support unit of the CC-Link IE Field Network cannot be described in version 1.1 or earlier. Specifications relating to this convention are mainly written in the following places.
5.3.1.3 COMM_IF_INFO parts description
(31) SafetyProtocolVersion element description - (35) SupportSafetyCommService element item description
5.5.5 Parts and elements showing parameters for safety communication function support
5.5.6 MESSAGE specifications for parameter writing of safety communication function support

(4) Description specifications for the Ethernet connection unit
Descriptions relating to the Ethernet connection unit cannot be described in version 1.1 or earlier. Specifications relating to this convention are mainly written in the following places.
5.3.1.3 COMM_IF_INFO parts description
(36) SupportIPAddressSize element item description - (48) SupportFlg_ReadStatus2 element item description
5.5.7 Conventions of parameters which can read and write using parameter IDs

9.3 Additional specifications for CSP+ specifications Version 2.1

If CSP+ specification version is written in 2.0 or earlier, description of the following conventions and specifications cannot be performed.

(1) Description specifications for the CC-Link IE Field Network Basic connection unit
Conventions relating to the CC-Link IE Field Network Basic connection unit cannot be described in version 2.0 or earlier. Specifications relating to this convention are mainly written in the following places.
4.2.7 CSP+ file structure of Ethernet connection unit
5.3.1.3 Description details of COMM_IF_INFO part

- Convention of the COMM_IF_INFO part when the communication interface is Ethernet
  (49) DevModel element description
  to (52) EthernetCommFunction element item description

(2) Description specifications for acquiring energy management information
Conventions for acquiring energy management information cannot be described in version 2.0 or earlier. Specifications relating to this convention are mainly written in the following places.
5.5.8 Parts and elements related to the acquisition of energy management information

(3) Description specifications for acquiring status
Conventions for acquiring the status code and displaying detailed information of the Ethernet connection unit cannot be described in version 2.0 or earlier. Specifications relating to this convention are mainly written in the following places.
5.5.9 Convention for acquiring status codes

9.4 Additional specifications for CSP+ specifications Version 2.2

If CSP+ specification version is written in 2.1 or earlier, description of the following conventions and specifications cannot be performed.

(1) Description specifications for acquiring maintenance information
Conventions for acquiring maintenance information cannot be described in version 2.1 or earlier. Specifications relating to this convention are mainly written in the following places.
4.3.1.21. MESSAGE_TYPE conventions
Addition of the type value "MAINTENANCE" and the detailed operating specifications of "MONITOR"
4.3.1.22. METHOD_TYPE conventions
Addition of the type value "MAINTENANCE" and the detailed operating specifications of "MONITOR"
4.3.1.32. RELATED_MESSAGE conventions
Addition of the keyword "POST"
4.3.1.33. RELATED_METHOD conventions
Addition of the keyword "POST"

9.5 Additional Specifications for CSP+ Specification Version 3.0

If CSP+ specification version is written with 2.2 or earlier, descriptions related to the following conventions and specifications are not permitted.

With a network connection device other than CC-Link IE TSN, create using the CSP+ specifications with version 2.2 or earlier.

(1) Description specifications related to CC-Link IE TSN connection unit

• Description specifications for CC-Link IE TSN connection unit

Conventions related to the CC-Link IE TSN connection unit and CC-Link IE TSN safety communication function support unit cannot be used in version 2.2 or earlier. Specifications related to these conventions are mainly written in the following.

4.1.1 CSP+ file name
4.1.2 Data configuration with compression management
4.2.8 CSP+ file structure of the CC-Link IE TSN connection unit
4.3.1.3 ASSIGN conventions
4.3.1.6 COMMENT conventions
4.3.1.21 MESSAGE_TYPE conventions
4.3.1.31 RELATED_ELE conventions
5.2.1 DEVICE_INFO part
5.3.1.1 Overview
5.3.1.3 Description of COMM_IF_INFO part
(53) Item description of StationMode element to (100) Item description of R_General_Address element
5.5.1 Elements related to system area
5.5.10.1 Parts and elements related to error related information
5.5.10.2 Convention related to MESSAGE for slave station parameter automatic setting
8 Description Specifications of Object Dictionary File

• Description specifications for CC-Link IE TSN safety communication support unit

4.2.9 In the case of the CC-Link IE TSN safety communication function support unit
5.3.1.3 Description of COMM_IF_INFO part
(101) Item description of SafetyProtocolVersion element to (112) Item description of R_Safe_Address element
5.5.10.3 Parts and elements indicating parameters for safety communication function
5.5.10.4 MESSAGE specifications for writing the safety communication function parameter

10. Related Specification Documents

[1] CC-Link Specification (Overview, Protocol) BAP-C2001-001
[2] CC-Link IE CC-Link Field Network Specification (Physical Layer and Data Link Layer) BAP-C2005-002
[3] CC-Link IE CC-Link Field Network Specification (Application Layer Service) BAP-C2005-003
[4] CC-Link IE CC-Link Field Network Specification (Application Layer Protocol) BAP-C2005-004
[5] CC-Link IE CC-Link Field Network Specification (Communication Profile) BAP-C2005-005
[6] CC-Link IE CC-Link Field Network Specification (Device Profile)BAP-C2005-007
[7] CC-Link IE Safety Communication Function Specification (Application Layer Service and Protocol) BAP-C2007ENG-002
[8] CC-Link IE Safety Communication Function Specification (Communication profile) BAP-C2007ENG-003
[9] SLMP (Seamless Message Protocol) Specification (Overview) BAP-C2006ENG-001
[10] SLMP (Seamless Message Protocol) Specification (Protocol) BAP-C2006ENG-003
[11] CC-Link IE Field Network Basic Specification BAP-C2010ENG-004
[12] CC-Link IE TSN Specifications (Overview) BAP-C2011ENG-001
[13] CC-Link IE TSN Specifications (Application Layer Protocol) BAP-C2011ENG-004
[14] CC-Link IE Safety Communication Function Specification (Overview) BAP-C2007ENG-001

CC-Link
CC-Link IE

Prepared in November, 2019