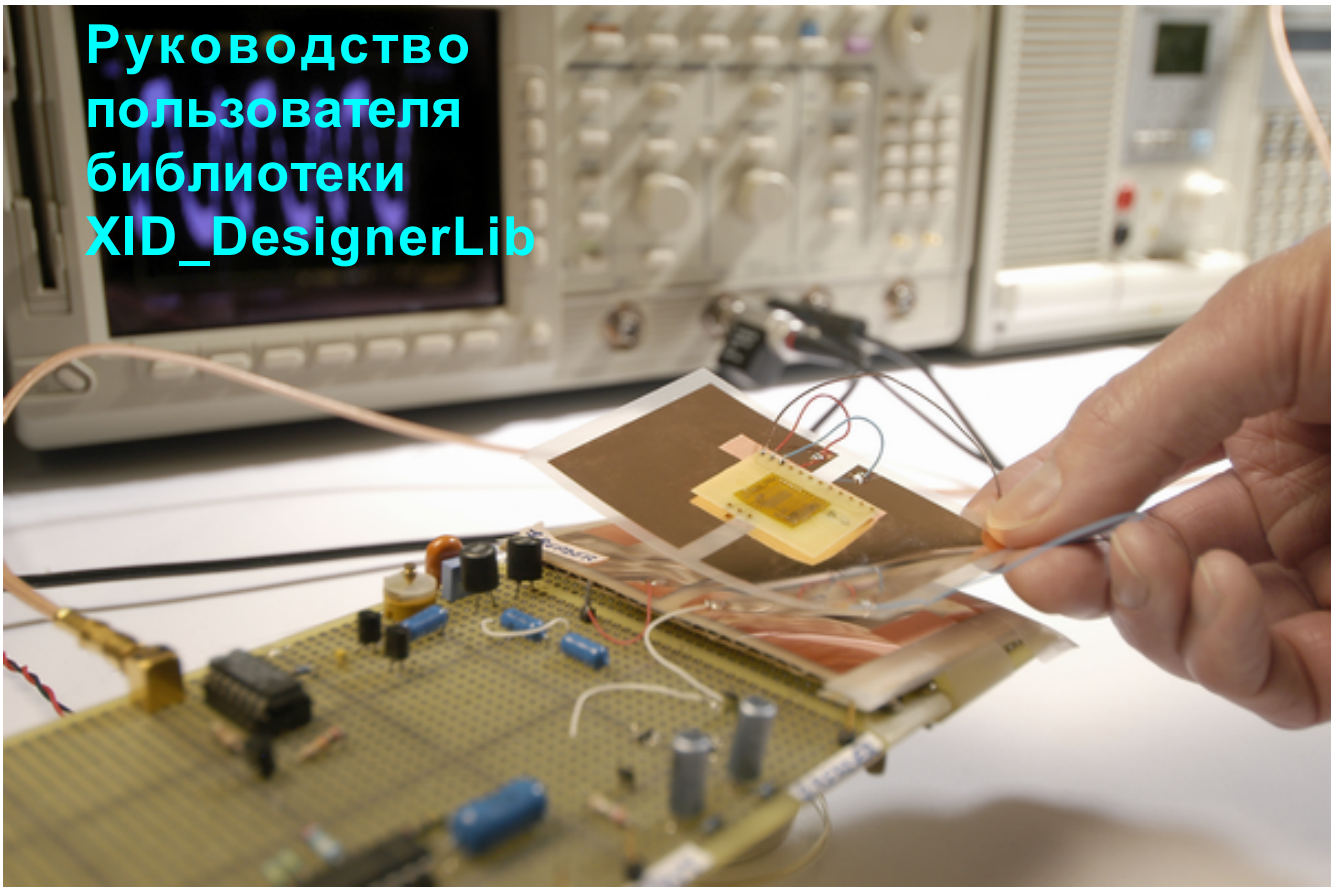


Руководство пользователя библиотеки XID_DesignerLib

Руководство пользователя библиотеки XID_DesignerLib

© 2011 MicroEM

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: Февраль 2011 in Russia, Zelenograd

Оглавление

Введение 0

Глава I Введение 32

1 О библиотеке ... 32

Глава II Установка библиотеки на систему 34

1 Установка пакета разработчика ... 34

2 Интеграция библиотеки в пакет установки приложения ... 34

Глава III Подключение библиотеки к наиболее распространенным типам проектов 37

1 MS Visual Studio 6.0 ... 37

Visual C++ 6.0 ... 37

Visual Basic 6.0 ... 37

2 MS Visual Studio 2003 ... 37

Visual C++ 2003 ... 37

Visual Basic 2003 ... 37

Visual C# 2003 ... 38

Visual J# 2003 ... 38

3 MS Visual Studio 2005/2008 ... 38

Visual C++ 2005/2008 ... 38

Visual Basic 2005/2008 ... 38

Visual C# 2005/2008 ... 39

Visual J# 2005/2008 ... 39

Visual C++ CLR 2005/2008 ... 39

4 Borland Delphi 7.0 ... 40

5 Borland C++ Builder 5.0 ... 40

Глава IV Порядок работы с объектами COM в наиболее распространенных типах проектов 42

1 Создание объектов ... 42

2 Доступ к методам объектов ... 44

3 Доступ к свойствам объектов ... 45

4 Доступ к интерфейсам объектов ... 46

5 Удаление объектов ... 47

6 Реализация событий объектов ... 49

Глава V Описание типов объектов библиотеки 53

1 Базовые объекты ... 53

XID_Manager ... 53

Методы ... 53

© 2011 MicroEM

DetectDevices ... 53
DetectVendorDevices ... 53
GetDevices ... 53
XID_DevicesList ... 54
Методы ... 54
Clear ... 54
Add ... 54
AddDevicesList ... 54
Remove ... 54
GetDevicesByClass ... 55
GetReaders ... 55
CopyTo ... 55
Свойства ... 56
Count ... 56
Item ... 56
GetDeviceIndex ... 56
XID_Device ... 56
Методы ... 56
CloneFrom ... 56
Свойства ... 57
ComPort ... 57
ID ... 57
BSA ... 57
Model ... 57
Version ... 57
Vendor ... 57
CommBaudRate ... 58
CommunicationType ... 58
DeviceClass ... 58
IsEqualTo ... 58
Типы ... 58
XID_eDeviceClass ... 58
XID_eCommBaudRate ... 59
XID_eCommunicationType ... 59
XID_ControlX ... 59
Методы ... 59
SelectDevice ... 59
LinkDevice ... 60
UnlinkDevice ... 60
LinkDevices ... 60
UnlinkAllDevices ... 60
Свойства ... 60
LastError ... 60
LastErrorDescription ... 61
Language ... 61
TimeOut ... 61
Типы ... 61
XID_eSlotErrorCodes ... 61
XID_eGeneralErrorCodes ... 61
XID_eCodedTimeSlots ... 62
XID_eLanguage ... 62
События ... 63
OnActivateISO14443A ... 63
OnActivateISO14443B ... 63

OnActivateISO15693 ... 63
Сложные функции (Complex Functions) ... 63
Full Anticollision ... 64
Свойства ... 64
Readers ... 64
Tags ... 64
in_Standards ... 64
UseEPC ... 64
UseISO14443A ... 65
UseISO14443B ... 65
UseISO15693 ... 65
UseICode1 ... 65
in_CodedTimeSlots ... 65
in_AutoSwitchRF ... 65
Методы ... 65
do_FullAnticollision ... 65
Read Data From Cards ... 66
Свойства ... 66
Readers ... 66
Tags ... 66
in_Standards ... 66
UseISO14443B ... 66
UseISO14443A ... 66
UseISO15693 ... 67
UseICode1 ... 67
UseEPC ... 67
in_CodedTimeSlots ... 67
in_AutoSwitchRF ... 67
Методы ... 67
do_ReadDataFromCards ... 67
Write Data To Cards ... 67
Свойства ... 68
Readers ... 68
Tags ... 68
in_Standards ... 68
UseISO15693 ... 68
UseISO14443A ... 68
UseISO14443B ... 68
UseEPC ... 69
UseICode1 ... 69
in_CodedTimeSlots ... 69
in_DataAsTextString ... 69
in_DataAsHexString ... 69
in_StartBlockShift ... 69
in_BlocksCount ... 69
in_BlockSize ... 69
in_AutoSwitchRF ... 70
Методы ... 70
do_WriteDataToCards ... 70
Типы ... 70
XID_eCodedTimeSlots ... 70
Функции командных запросов (Descrete Query Functions) ... 70
Общие функции устройств (Any Device Functions) ... 71
ReadConfig ... 71

© 2011 MicroEM

Свойства ... 71
in_ConfigLength ... 71
out_ConfigData ... 71
Методы ... 71
do_ReadConfig ... 71
WriteConfig ... 71
Свойства ... 71
in_ConfigLength ... 71
in_ConfigData ... 72
Методы ... 72
do_WriteConfig ... 72
ReadBusAddress ... 72
Свойства ... 72
out_BusAddress ... 72
Методы ... 72
do_ReadBusAddress ... 72
WriteBusAddress ... 72
Свойства ... 73
in_BusAddress ... 73
Методы ... 73
do_WriteBusAddress ... 73
Signal ... 73
Свойства ... 73
in_SignalType ... 73
in_SignalLength ... 73
in_SignalsCount ... 73
in_LedColor ... 73
in_LedBlink ... 73
in_RemainingColor ... 74
Методы ... 74
do_Signal ... 74
UpdateFirmware ... 74
Свойства ... 74
in_FirmwareFileName ... 74
in_RawUpdate ... 74
Методы ... 74
do_UpdateFirmware ... 74
Типы ... 74
XID_eSignalTypes ... 74
XID_eLedColor ... 75
Функции считывателя (Reader Functions) ... 75
ResetRF ... 75
Свойства ... 75
in_ResetTime ... 75
Методы ... 75
do_ResetRF ... 75
RF_ON ... 75
Методы ... 76
do_RF_ON ... 76
RF_OFF ... 76
Методы ... 76
do_RF_OFF ... 76
GetKeysState ... 76
Свойства ... 76

out_KeysState ... 76
Методы ... 76
do_GetKeysState ... 76
SetSignalState ... 77
Свойства ... 77
in_New SignalState ... 77
Методы ... 77
do_SetSignalState ... 77
GetNoiseFactor ... 77
Свойства ... 77
out_NoiseFactor ... 77
Методы ... 77
do_GetNoiseFactor ... 77
Функции по стандартам меток (Functions By Tags Standards) ... 78
EPC Global ... 78
Generation I ... 78
Class 1 ... 78
HF(13.56MHz) ... 78
Anticollision ... 79
Свойства ... 79
in_HashValue ... 79
in_CodedTimeSlots ... 79
in_Mask ... 79
in_MaskLengthInBits ... 79
out_TimeSlots ... 79
out_TimeSlotsStatuses ... 79
out_EPCs ... 79
Методы ... 80
do_Anticollision ... 80
Write Block ... 80
Свойства ... 80
in_BlockAddress ... 80
in_Block ... 80
Методы ... 80
do_WriteBlock ... 80
Destroy ... 80
Свойства ... 80
in_EPC ... 80
in_DestroyCode ... 81
Методы ... 81
do_Destroy ... 81
ISO ... 81
ISO15693 ... 81
Anticollision ... 81
Свойства ... 81
in_Use1TimeSlot ... 81
in_AddAFI ... 82
in_AFI ... 82
in_MaskLengthInBits ... 82
in_Mask ... 82
out_TimeSlots ... 82
out_arrTimeSlotsStatuses ... 82
out_TimeSlotsStatuses ... 82
out_arrUIDs ... 82

© 2011 MicroEM

out_UIDs ... 83
Методы ... 83
do_Anticollision ... 83
GetBlocksSecurities ... 83
Свойства ... 83
in_SelectionType ... 83
in_UID ... 83
in_BlockAddress ... 83
in_BlocksCount ... 83
out_BlocksSecurities ... 83
Методы ... 84
do_GetBlocksSecurities ... 84
GetSystemInfo ... 84
Свойства ... 84
in_SelectionType ... 84
in_UID ... 84
out_SystemInformation ... 84
Свойства ... 84
SupportAFI ... 84
SupportDSFID ... 84
HasMemSizeInfo ... 85
HasICreferenceInfo ... 85
Raw InfoAsHexText ... 85
AFI ... 85
DSFID ... 85
MemoryInfo ... 85
Свойства ... 85
BlockSize ... 85
NumberOfBlocks ... 85
IC_ReferenceInfo ... 86
Методы ... 86
do_GetSystemInfo ... 86
LockAFI ... 86
Свойства ... 86
in_SelectionType ... 86
in_UID ... 86
in_WaitForEOF ... 86
Методы ... 86
do_LockAFI ... 86
LockBlock ... 86
Свойства ... 87
in_SelectionType ... 87
in_UID ... 87
in_WaitForEOF ... 87
in_BlockAddress ... 87
Методы ... 87
do_LockBlock ... 87
LockDSFID ... 87
Свойства ... 87
in_SelectionType ... 87
in_UID ... 88
in_WaitForEOF ... 88
Методы ... 88
do_LockDSFID ... 88

ReadBlock ... 88
Свойства ... 88
in_AddSecurityByte ... 88
in_SelectionType ... 88
in_UID ... 88
in_BlockAddress ... 89
out_BlockSize ... 89
out_BlockAsHexString ... 89
out_BlockAsTextString ... 89
out_BlockSecurity ... 89
Методы ... 89
do_ReadBlock ... 89
ReadBlocks ... 89
Свойства ... 90
in_AddSecurityByte ... 90
in_SelectionType ... 90
in_UID ... 90
in_BlockAddress ... 90
in_BlocksCount ... 90
out_BlockSize ... 90
out_arrBlocksAsHexStrings ... 90
out_BlocksAsHexStrings ... 90
out_arrBlocksAsTextStrings ... 91
out_BlocksAsTextStrings ... 91
out_BlocksSecurities ... 91
Методы ... 91
do_ReadBlocks ... 91
ResetToReady ... 91
Свойства ... 91
in_SelectionType ... 91
in_UID ... 91
Методы ... 92
do_ResetToReady ... 92
Select ... 92
Свойства ... 92
in_UID ... 92
Методы ... 92
do_Select ... 92
StayQuiet ... 92
Свойства ... 92
in_UID ... 92
Методы ... 93
do_StayQuiet ... 93
WriteAFI ... 93
Свойства ... 93
in_SelectionType ... 93
in_UID ... 93
in_WaitForEOF ... 93
in_AFI ... 93
Методы ... 93
do_WriteAFI ... 93
WriteBlock ... 93
Свойства ... 94
in_SelectionType ... 94

© 2011 MicroEM

in_UID ... 94
in_WaitForEOF ... 94
in_BlockAddress ... 94
in_BlockSize ... 94
in_BlockAsHexString ... 94
in_BlockAsTextString ... 94
Методы ... 95
do_WriteBlock ... 95
WriteBlocks ... 95
Свойства ... 95
in_SelectionType ... 95
in_UID ... 95
in_WaitForEOF ... 95
in_BlockAddress ... 95
in_BlocksCount ... 95
in_BlockSize ... 96
in_BlocksAsHexStrings ... 96
in_BlocksAsTextStrings ... 96
Методы ... 96
do_WriteBlocks ... 96
WriteDSFID ... 96
Свойства ... 96
in_SelectionType ... 96
in_UID ... 96
in_WaitForEOF ... 97
in_DSFID ... 97
Методы ... 97
do_WriteDSFID ... 97
Типы ... 97
XID_eIS015693_QueryTypes ... 97
IXID_ISO15693_Security ... 97
Свойства ... 97
Raw Byte ... 97
Locked ... 97
ISO14443A ... 98
ActivateIdle ... 98
Свойства ... 98
out_UID ... 98
out_Is_ISO14443_4_Compatible ... 98
Методы ... 98
do_ActivateIdle ... 98
Halt ... 98
Методы ... 98
do_Halt ... 98
WakeUp ... 99
Свойства ... 99
in_UID ... 99
Out_Is_ISO14443_4_Compatible ... 99
Методы ... 99
do_WakeUp ... 99
ISO14443B ... 99
ActivateIdle ... 99
Свойства ... 100
in_AppFamilyID ... 100

in_SlotsCount ... 100
out_CardsCount ... 100
out_GetCardInfo ... 100
Методы ... 100
do_ActivateIdle ... 100
Attrib ... 100
Свойства ... 100
in_PUPI ... 100
in_Param1 ... 101
MinimumTR0 ... 101
MinimumTR1 ... 101
RequiredSOF ... 101
RequiredEOF ... 101
in_Param2 ... 101
MaxFrameSize ... 102
PCD_to_PICC_BitRate ... 102
PICC_to_PCD_BitRate ... 102
in_Param3 ... 102
ProtocolType ... 102
in_Param4 ... 102
CardIdentifier ... 102
inout_arrHigerLevelINF ... 102
out_MaxBufLenIndex ... 103
out_CardID ... 103
Методы ... 103
do_Attrib ... 103
Halt ... 103
Свойства ... 103
in_PUPI ... 103
Методы ... 104
do_Halt ... 104
WakeUp ... 104
Свойства ... 104
in_AppFamilyID ... 104
in_SlotsCount ... 104
out_CardsCount ... 104
out_GetCardInfo ... 104
Методы ... 104
do_WakeUp ... 104
Типы ... 104
XID_eISO14443B_SlotsCount ... 104
XID_eISO14443B_MinimumTR0 ... 105
XID_eISO14443B_MinimumTR1 ... 105
XID_eISO14443B_MaxFrameSize ... 105
XID_eISO14443B_BitRate ... 106
XID_eISO14443B_ProtocolType ... 106
IXID_ISO14443B_Card ... 106
ProtocolInfo ... 107
ApplicationData ... 107
PUPI ... 107
NXP ... 107
ICode ... 107
ICode1 ... 108
Anticollision ... 108

© 2011 MicroEM

Свойства ... 108
in_HashValue ... 108
in_FamilyCode ... 108
in_ApplicationIdentifier ... 108
in_CodedTimeSlots ... 108
out_TimeSlots ... 108
out_TimeSlotsStatuses ... 109
out_IDs ... 109
Методы ... 109
do_Anticollision ... 109
EAS ... 109
Свойства ... 109
in_FamilyCode ... 109
in_ApplicationIdentifier ... 109
Методы ... 109
do_EAS ... 109
ReadSelected ... 109
Свойства ... 110
in_BlockAddress ... 110
in_BlocksCount ... 110
out_TimeSlots ... 110
out_TimeSlotsStatuses ... 110
out_BlocksAsTextStrings ... 110
out_BlocksAsHexStrings ... 110
Методы ... 110
do_ReadSelected ... 110
ReadUnselected ... 111
Свойства ... 111
in_HashValue ... 111
in_FamilyCode ... 111
in_ApplicationIdentifier ... 111
in_CodedTimeSlots ... 111
in_BlockAddress ... 111
in_BlocksCount ... 111
out_TimeSlots ... 111
out_TimeSlotsStatuses ... 112
out_BlocksAsTextString ... 112
out_BlocksAsHexString ... 112
Методы ... 112
do_ReadUnselected ... 112
ResetQuietBit ... 112
Методы ... 112
do_ResetQuietBit ... 112
WriteBlock ... 112
Свойства ... 113
in_HashValue ... 113
in_BlockAddress ... 113
in_SelectTimeSlot ... 113
in_BlockAsTextString ... 113
in_BlockAsHexString ... 113
out_TimeSlots ... 113
out_TimeSlotsStatuses ... 113
out_IDs ... 113
Методы ... 114

do_WriteBlock ... 114
Halt ... 114
Свойства ... 114
in_HashValue ... 114
in_SelectTimeSlot ... 114
out_TimeSlots ... 114
out_TimeSlotsStatuses ... 114
out_IDs ... 114
Методы ... 115
do_Halt ... 115
ICode2(SLI) ... 115
ISO15693 ... 115
SetEAS ... 115
Свойства ... 115
in_SelectionType ... 115
in_UID ... 115
in_ManufacturerCode ... 116
Методы ... 116
do_SetEAS ... 116
ResetEAS ... 116
Свойства ... 116
in_SelectionType ... 116
in_UID ... 116
in_ManufacturerCode ... 116
Методы ... 116
do_ResetEAS ... 116
LockEAS ... 116
Свойства ... 117
in_SelectionType ... 117
in_UID ... 117
in_ManufacturerCode ... 117
Методы ... 117
do_LockEAS ... 117
EASAlarm ... 117
Свойства ... 117
in_SelectionType ... 117
in_UID ... 117
in_ManufacturerCode ... 118
out_EASSequenceAsHexString ... 118
Методы ... 118
do_EASAlarm ... 118
ICode EPC Gen I Class 1 13.56MHz ... 118
Mifare ... 118
Standard 1KB, 4KB Memory ... 118
ISO14443A ... 119
Authenticate ... 119
Свойства ... 119
in_UID ... 119
in_KeyType ... 119
in_KeyNumber ... 119
in_SectorNumber ... 119
in_CodedKey ... 120
Методы ... 120
do_Authenticate ... 120

© 2011 MicroEM

do_AuthenticateWithKey ... 120
LoadKey ... 120
Свойства ... 120
in_KeyType ... 120
in_KeyNumber ... 120
in_SectorNumber ... 120
in_Key ... 121
Методы ... 121
do_LoadKey ... 121
GenerateKey ... 121
Свойства ... 121
out_Key ... 121
Методы ... 121
do_GenerateKey ... 121
CodeKey ... 121
Свойства ... 121
in_Key ... 121
out_CodedKey ... 122
Методы ... 122
do_CodeKey ... 122
ReadBlock ... 122
Свойства ... 122
in_BlockNumber ... 122
out_BlockAsHexString ... 122
out_BlockAsTextString ... 122
Методы ... 122
do_ReadBlock ... 122
WriteBlock ... 123
Свойства ... 123
in_BlockNumber ... 123
in_BlockAsHexString ... 123
in_BlockAsTextString ... 123
Методы ... 123
do_WriteBlock ... 123
ReadTrailer ... 123
Свойства ... 123
out_KeyA ... 123
out_KeyB ... 124
out_Byte9 ... 124
out_AccessConditions ... 124
Методы ... 124
do_ReadTrailer ... 124
WriteTrailer ... 124
Свойства ... 124
in_KeyA ... 124
in_KeyB ... 124
in_Byte9 ... 124
in_AccessConditions ... 125
Методы ... 125
do_WriteTrailer ... 125
ReadValue ... 125
Свойства ... 125
in_BlockNumber ... 125
out_Value ... 125

out_Address ... 125
Методы ... 125
do_ReadValue ... 125
WriteValue ... 125
Свойства ... 126
in_BlockNumber ... 126
in_Value ... 126
in_Address ... 126
Методы ... 126
do_WriteValue ... 126
IncrementValue ... 126
Свойства ... 126
in_SourceBlockAddress ... 126
in_AddendValue ... 126
in_DestinationBlockNumber ... 127
Методы ... 127
do_IncrementValue ... 127
DecrementValue ... 127
Свойства ... 127
in_SourceBlockAddress ... 127
in_SubtrahendValue ... 127
in_DestinationBlockNumber ... 127
Методы ... 127
do_DecrementValue ... 127
CopyValue ... 127
Свойства ... 128
in_SourceBlockAddress ... 128
in_DestinationBlockNumber ... 128
Методы ... 128
do_CopyValue ... 128
Типы ... 128
IXID_MifareStandard_AccessConditions ... 128
Свойства ... 128
OfBlock ... 128
Свойства ... 128
AccessConditionType ... 128
WriteData ... 128
ReadData ... 128
IncrementValue ... 129
DecrementValue ... 129
CopyValue ... 129
Типы ... 129
XID_eMifareStandardAccessType ... 129
XID_eMifareStandardAccessConditionType ... 129
OfSectorTrailer ... 130
Свойства ... 130
AccessConditionType ... 130
WriteKeyA ... 130
ReadKeyA ... 130
WriteAccessBits ... 130
ReadAccessBits ... 130
WriteKeyB ... 130
ReadKeyB ... 130
XID_eMifareStandardKeyType ... 131

© 2011 MicroEM

Ultra Light ... 131
ISO14443A ... 131
ReadAllPages ... 131
Свойства ... 131
out_Pages ... 131
Методы ... 132
do_ReadAllPages ... 132
ReadPages ... 132
Свойства ... 132
in_StartPageNumber ... 132
out_Pages ... 132
Методы ... 132
do_ReadPages ... 132
WritePage ... 132
Свойства ... 132
in_PageNumber ... 132
in_Pages ... 133
Методы ... 133
do_WritePage ... 133
WriteAllPages ... 133
Свойства ... 133
in_Pages ... 133
Методы ... 133
do_WriteAllPages ... 133
ReadLock ... 133
Свойства ... 133
out_Pages ... 133
Методы ... 134
do_ReadLock ... 134
WriteLock ... 134
Свойства ... 134
in_Pages ... 134
Методы ... 134
do_WriteLock ... 134
ReadOTP ... 134
Свойства ... 134
out_Pages ... 134
Методы ... 135
do_ReadOTP ... 135
WriteOTP ... 135
Свойства ... 135
in_Pages ... 135
Методы ... 135
do_WriteOTP ... 135
Типы ... 135
IXID_MifareUL_Pages ... 135
Свойства ... 135
SerialNumber ... 135
SerialNumberIsOK ... 135
LockInfo ... 135
PageLocked ... 136
OTP_Locked ... 136
LockOfOTP_Blocked ... 136
LockOfPages4to9_Blocked ... 136

LockOfPages10to15_Blocked ... 136
OTP ... 136
LockBlocked ... 136
Locked ... 136
Bits ... 137
Value ... 137
PagesArray ... 137
AllPagesAsHexString ... 137
PagesHexStringArray ... 137
Методы ... 137
GetDataPage ... 137
DataAsTextString ... 137
DataAsHexString ... 137
DataAsValue ... 138
Bytes ... 138
Locked ... 138
GetLockBlocked ... 138
Des Fire (EV1) ... 138
ISO14443A ... 138
RequestAnsw erToSelect ... 138
Свойства ... 139
out_CardNumber ... 139
Методы ... 139
do_RequestAnsw erToSelect ... 139
Authenticate ... 139
Свойства ... 139
in_KeyNumber ... 139
in_AuthenticationType ... 139
in_KeyAsHexString ... 139
Методы ... 139
do_Authenticate ... 139
SetConfiguration ... 140
Свойства ... 140
in_ConfigurationOption ... 140
in_FormatCardDisabled ... 140
in_RandomIDEnabled ... 140
in_Key ... 140
in_KeyVersion ... 140
in_ATS ... 140
Методы ... 140
do_SetConfiguration ... 140
FreeMem ... 141
Свойства ... 141
out_MemorySize ... 141
Методы ... 141
do_FreeMem ... 141
ChangeKeySettings ... 141
Свойства ... 141
in_PiccMasterKeySettings ... 141
in_ApplicationMasterKeySettings ... 141
Методы ... 141
do_ChangeKeySettings ... 141
GetMasterKeySettings ... 142
Свойства ... 142

© 2011 MicroEM

out_MasterKeySettings ... 142
out_KeySettings ... 142
Методы ... 142
do_GetKeySettings ... 142
ChangeKey ... 142
Свойства ... 142
in_KeyVersion ... 142
in_Key ... 142
in_KeyNumber ... 142
in_ChangeKeyType ... 143
Методы ... 143
do_ChangeKey ... 143
GetKeyVersion ... 143
Свойства ... 143
in_KeyNumber ... 143
out_KeyVersion ... 143
Методы ... 143
do_GetKeyVersion ... 143
GetCardUID ... 143
Свойства ... 144
out_CardUID ... 144
Методы ... 144
do_GetCardUID ... 144
GetDFNames ... 144
Свойства ... 144
out_DFNames ... 144
Методы ... 144
do_GetDFNames ... 144
FormatPICC ... 144
Методы ... 144
do_FormatPICC ... 144
GetVersion ... 145
Свойства ... 145
out_Version ... 145
Методы ... 145
do_GetVersion ... 145
Типы ... 145
IXID_MifareDESfire_Version ... 145
Свойства ... 145
HardwareInfo ... 145
Свойства ... 145
HardwareVendorID ... 145
HardwareType ... 145
HardwareSubtype ... 145
HardwareMajorVersionNumber ... 146
HardwareMinorVersionNumber ... 146
HardwareStorageSize ... 146
HardwareCommunicationProtocol ... 146
SoftwareInfo ... 146
Свойства ... 146
SoftwareVendorID ... 146
SoftwareType ... 146
SoftwareSubtype ... 147
SoftwareMajorVersionNumber ... 147

SoftwareMinorVersionNumber ... 147
SoftwareStorageSize ... 147
SoftwareCommunicationProtocol ... 147
ProductionButchNumber ... 147
SerialNumber ... 147
ProductionWeek ... 147
ProductionYear ... 148
Свойства ... 148
LastError ... 148
LastErrorDescription ... 148
GetNewCardNumber ... 148
Applications ... 148
Методы ... 148
FreeCardNumber ... 148
Типы ... 148
IXID_MifareDESfire_Applications ... 148
CreateApplication ... 148
Свойства ... 149
in_ApplicationID ... 149
in_ApplicationMasterKeySettings ... 149
in_ApplicationKeySettings ... 149
Методы ... 149
do_CreateApplication ... 149
SelectApplication ... 149
Свойства ... 149
in_ApplicationID ... 149
Методы ... 149
do_SelectApplication ... 149
DeleteApplication ... 149
Свойства ... 150
in_ApplicationID ... 150
Методы ... 150
do_DeleteApplication ... 150
CommitTransaction ... 150
Методы ... 150
do_CommitTransaction ... 150
AbortTransaction ... 150
Методы ... 150
do_AbortTranscation ... 150
GetApplicationIDs ... 150
Методы ... 150
do_GetApplicationIDs ... 150
GetKeySettings ... 150
Свойства ... 151
out_MasterKeySettings ... 151
out_MasterKeyType ... 151
out_ApplicationMasterKeySettings ... 151
out_ApplicationKeySettings ... 151
Методы ... 151
do_GetKeySettings ... 151
Свойства ... 151
Count ... 151
Item ... 151
Типы ... 152

© 2011 MicroEM

IXID_MifareDESfire_Application ... 152
GetKeySettings ... 152
Свойства ... 152
out_ApplicationMasterKeySettings ... 152
out_ApplicationKeySettings ... 152
Методы ... 152
do_GetKeySettings ... 152
Свойства ... 152
D ... 152
Files ... 152
Методы ... 152
Delete ... 152
Select ... 152
CommitTransaction ... 152
AbortTransaction ... 152
Типы ... 153
IXID_MifareDESfire_Files ... 153
GetFileIDs ... 153
Методы ... 153
do_GetFileIDs ... 153
GetISOFileIDs ... 153
Методы ... 153
do_GetISOFileIDs ... 153
GetFileSettings ... 153
Свойства ... 153
in_FileID ... 153
out_FileType ... 153
out_CommunicationSettings ... 153
out_ReadAccess ... 154
out_WriteAccess ... 154
out_ReadAndWriteAccess ... 154
out_ChangeAccessRights ... 154
out_FileSize ... 154
out_LowerLimit ... 154
out_UpperLimit ... 154
out_LimitedCreditValue ... 154
out_LimitedCreditEnabled ... 154
out_RecordSize ... 155
out_MaximumNumberOfRecords ... 155
out_CurrentNumberOfRecords ... 155
Методы ... 155
do_GetFileSettings ... 155
ChangeFileSettings ... 155
Свойства ... 155
in_FileID ... 155
in_CommunicationSettings ... 155
in_ReadAccess ... 155
in_WriteAccess ... 156
in_ReadAndWriteAccess ... 156
in_ChangeAccessRights ... 156
Методы ... 156
do_ChangeFileSettings ... 156
CreateStdDataFile ... 156
Свойства ... 156

in_FileID ... 156
in_CommunicationSettings ... 156
in_ReadAccess ... 156
in_WriteAccess ... 156
in_ReadAndWriteAccess ... 157
in_ChangeAccessRights ... 157
in_FileSize ... 157
Методы ... 157
do_CreateStdDataFile ... 157
CreateBackupDataFile ... 157
Свойства ... 157
in_FileID ... 157
in_CommunicationSettings ... 157
in_ReadAccess ... 157
in_WriteAccess ... 157
in_ReadAndWriteAccess ... 158
in_ChangeAccessRights ... 158
in_FileSize ... 158
Методы ... 158
do_CreateBackupDataFile ... 158
CreateValueFile ... 158
Свойства ... 158
in_FileID ... 158
in_CommunicationSettings ... 158
in_ReadAccess ... 158
in_WriteAccess ... 159
in_ReadAndWriteAccess ... 159
in_ChangeAccessRights ... 159
in_LowerLimit ... 159
in_UpperLimit ... 159
in_Value ... 159
in_LimitedCreditEnabled ... 159
Методы ... 159
do_CreateValueFile ... 159
CreateLinearRecordFile ... 159
Свойства ... 160
in_FileID ... 160
in_CommunicationSettings ... 160
in_ReadAccess ... 160
in_WriteAccess ... 160
in_ReadAndWriteAccess ... 160
in_ChangeAccessRights ... 160
in_RecordSize ... 160
in_MaxNumberOfRecords ... 160
Методы ... 161
do_CreateLinearRecordFile ... 161
CreateCyclicRecordFile ... 161
Свойства ... 161
in_FileID ... 161
in_CommunicationSettings ... 161
in_ReadAccess ... 161
in_WriteAccess ... 161
in_ReadAndWriteAccess ... 161
in_ChangeAccessRights ... 161

© 2011 MicroEM

in_RecordSize ... 161
in_MaxNumberOfRecords ... 162
Методы ... 162
do_CreateCyclicRecordFile ... 162
DeleteFile ... 162
Свойства ... 162
in_FileID ... 162
Методы ... 162
do_DeleteFile ... 162
ReadData ... 162
Свойства ... 162
in_FileID ... 162
in_CommunicationSettings ... 162
in_Offset ... 163
in_Length ... 163
out_DataAsHexString ... 163
Методы ... 163
do_ReadData ... 163
WriteData ... 163
Свойства ... 163
in_FileID ... 163
in_CommunicationSettings ... 163
in_Offset ... 163
in_Length ... 164
in_DataAsHexString ... 164
Методы ... 164
do_WriteData ... 164
GetValue ... 164
Свойства ... 164
in_FileID ... 164
in_CommunicationSettings ... 164
out_Value ... 164
Методы ... 164
do_GetValue ... 164
Credit ... 164
Свойства ... 165
in_FileID ... 165
in_CommunicationSettings ... 165
in_Value ... 165
Методы ... 165
do_Credit ... 165
Debit ... 165
Свойства ... 165
in_FileID ... 165
in_CommunicationSettings ... 165
in_Value ... 165
Методы ... 166
do_Debit ... 166
LimitedCredit ... 166
Свойства ... 166
in_FileID ... 166
in_CommunicationSettings ... 166
in_Value ... 166
Методы ... 166

do_LimitedCredit ... 166
WriteRecord ... 166
Свойства ... 166
in_FileID ... 166
in_CommunicationSettings ... 167
in_Offset ... 167
in_Length ... 167
in_DataAsHexString ... 167
Методы ... 167
do_WriteRecord ... 167
ClearRecordFile ... 167
Свойства ... 167
in_FileID ... 167
Методы ... 167
do_ClearRecordFile ... 167
ReadRecords ... 167
Свойства ... 168
in_FileID ... 168
in_CommunicationSettings ... 168
in_Offset ... 168
in_Length ... 168
out_DataAsHexString ... 168
Методы ... 168
do_ReadRecords ... 168
Свойства ... 168
Count ... 168
Item ... 168
Типы ... 169
XID_eMifareDESfire_CodingOfFileTypes ... 169
XID_eMifareDESfire_CodingOfCommunicationSettings ... 169
XID_eMifareDESfire_CodingOfAccessRights ... 169
IXID_MifareDESfire_File ... 170
GetSettings ... 170
Свойства ... 170
out_FileType ... 170
out_CommunicationSettings ... 170
out_ReadAccess ... 170
out_WriteAccess ... 170
out_ReadAndWriteAccess ... 170
out_ChangeAccessRights ... 170
out_FileSize ... 171
out_LowerLimit ... 171
out_UpperLimit ... 171
out_LimitedCreditValue ... 171
out_LimitedCreditEnabled ... 171
out_RecordSize ... 171
out_MaximumNumberOfRecords ... 171
out_CurrentNumberOfRecords ... 171
Методы ... 171
do_GetSettings ... 171
ChangeSettings ... 172
Свойства ... 172
in_CommunicationSettings ... 172
in_ReadAccess ... 172

© 2011 MicroEM

in_WriteAccess ... 172
in_ReadAndWriteAccess ... 172
in_ChangeAccessRights ... 172
Методы ... 172
do_ChangeSettings ... 172
ReadData ... 172
Свойства ... 173
in_Offset ... 173
in_Length ... 173
out_DataAsHexString ... 173
Методы ... 173
do_ReadData ... 173
WriteData ... 173
Свойства ... 173
in_Offset ... 173
in_Length ... 173
in_DataAsHexString ... 173
Методы ... 174
do_WriteData ... 174
GetValue ... 174
Свойства ... 174
out_Value ... 174
Методы ... 174
do_GetValue ... 174
Credit ... 174
Свойства ... 174
in_Value ... 174
Методы ... 174
do_Credit ... 174
Debit ... 174
Свойства ... 174
in_Value ... 174
Методы ... 175
do_Debit ... 175
LimitedCredit ... 175
Свойства ... 175
in_Value ... 175
Методы ... 175
do_LimitedCredit ... 175
WriteRecord ... 175
Свойства ... 175
in_Offset ... 175
in_Length ... 175
in_DataAsHexString ... 175
Методы ... 176
do_WriteRecord ... 176
ClearRecordFile ... 176
Методы ... 176
do_ClearRecordFile ... 176
ReadRecords ... 176
Свойства ... 176
in_Offset ... 176
in_Length ... 176
out_DataAsHexString ... 176

Методы ... 176
do_ReadRecords ... 176
Свойства ... 176
ID ... 176
ISOID ... 177
FileType ... 177
CommunicationSettings ... 177
ReadAccess ... 177
WriteAccess ... 177
ReadAndWriteAccess ... 177
ChangeAccessRights ... 177
Методы ... 177
Delete ... 177
IXID_MifareDESfire_ApplicationKeySettings ... 177
Свойства ... 177
MaximumNumberOfKeys ... 177
UsageOf2ByteFileIdentifiers ... 178
ApplicationCryptoMethod ... 178
Типы ... 178
XID_eMifareDESfire_UsageOf2ByteFileIdentifiers ... 178
XID_eMifareDESfire_ApplicationCryptoMethod ... 178
IXID_MifareDESfire_ApplicationMasterKeySettings ... 178
Свойства ... 178
AccessRightsForChangingApplicationKeys ... 178
AccessToApplicationMasterKeySettingsChanging ... 178
CreateAndDeleteFileAccessByApplicationMasterKey ... 179
FileDirectoryAccessByApplicationMasterKey ... 179
ApplicationMasterKeyChangeability ... 179
Типы ... 179
XID_eMifareDESfire_AccessRightsForChangingApplicationKeys ... 179

XID_eMifareDESfire_AccessToApplicationMasterKeySettingsChan
ging ... 180

XID_eMifareDESfire_CreateAndDeleteFileAccessByApplicationMa
sterKey ... 180
XID_eMifareDESfire_FileDirectoryAccessByApplicationMasterKey ... 180
XID_eMifareDESfire_ApplicationMasterKeyChangeability ... 181
XID_eMifareDESfire_PiccMasterKeyType ... 181
IXID_MifareDESfire_DFNames ... 181
Свойства ... 181
Count ... 181
Item ... 181
Типы ... 181
IXID_MifareDESfire_DFName ... 181
Свойства ... 181
AID ... 181
FID ... 182
DFName ... 182
IXID_MifareDESfire_KeySettings ... 182
Свойства ... 182
MaximumNumberOfKeys ... 182
UsageOf2ByteFileIdentifiers ... 182
ApplicationCryptoMethod ... 182
Типы ... 182

© 2011 MicroEM

XID_eMifareDESfire_UsageOf2ByteFileIdentifiers ... 182
XID_eMifareDESfire_ApplicationCryptoMethod ... 182
IXID_MifareDESfire_PrimaryKeySettings ... 183
Свойства ... 183
AccessToMasterKeySettingsChanging ... 183
CreateAndDeleteApplicationAccessByMasterKey ... 183
ApplicationDirectoryAccessByMasterKey ... 183
MasterKeyChangeability ... 183
Типы ... 183
XID_eMifareDESfire_AccessToMasterKeySettingsChanging ... 183
XID_eMifareDESfire_CreateAndDeleteApplicationAccessByMaster Key ... 183
XID_eMifareDESfire_ApplicationDirectoryAccessByMasterKey ... 184
XID_eMifareDESfire_MasterKeyChangeability ... 184
XID_eMifareDESfire_ErrorCodes ... 184
XID_eMifareDESfire_AuthenticationType ... 185
XID_eMifareDESfire_ConfigurationOptions ... 185
XID_TagsList ... 185
Методы ... 185
Clear ... 185
Add ... 186
AddTagsList ... 186
SubstractTagsList ... 186
CopyTo ... 186
Remove ... 187
DeleteTagsWithID ... 187
GetTagsByType ... 187
MutualExcludeDuplicates ... 187
SortByType ... 188
SetTextStringToAllTags ... 188
SetHexStringToAllTags ... 188
Свойства ... 188
Count ... 188
Item ... 188
GetTagIndex ... 189
XID_Tag ... 189
Методы ... 189
CloneFrom ... 189
Свойства ... 189
D ... 189
Type ... 189
TypeStr ... 190
Vendor ... 190
LastOperationStatus ... 190
BlockSize ... 190
BlocksCount ... 190
LastReader ... 190
GetSector ... 191
GetBlock ... 191
DataAsHexString ... 191
DataAsTextString ... 191
DataAsHexString ... 191
DataAsTextString ... 191

BlocksPerSector ... 192
SectorsCount ... 192
AllDataAsHexString ... 192
AllDataAsTextString ... 192
IsEqualTo ... 192
Типы ... 192
XID_eTagType ... 192
XID_eSlotErrorCodes ... 193
2 Упрощенные объекты ... 193
XID_SimpleReadersControl ... 193
Свойства ... 194
Language ... 194
LastError ... 194
LastErrorDescription ... 194
Методы ... 194
LinkReader ... 194
LinkReaders ... 194
UnlinkReader ... 195
UnlinkAllReaders ... 195
Beep ... 195
Blink ... 195
DetectReaders ... 195
DetectVendorReaders ... 196
Типы ... 196
XID_eLanguage ... 196
XID_eGeneralErrorCodes ... 196
События ... 197
OnTagEvent ... 197
OnSessionPaused ... 197
OnSessionStopped ... 197
OnActivateISO14443A ... 197
OnActivateISO14443B ... 198
OnActivateISO15693 ... 198
FindCards ... 198
Свойства ... 199
ExpectedCardsCount ... 199
UsedReaders ... 199
ExpectedCardsStandards ... 199
UseICode1 ... 199
UseISO15693 ... 199
UseEPC_G1C1HF ... 199
UseISO14443A ... 199
UseISO14443B ... 200
Методы ... 200
Execute ... 200
ReadData ... 200
Свойства ... 200
UsedReaders ... 200
ExpectedCardsStandards ... 201
Свойства ... 201
UseICode1 ... 201
UseISO15693 ... 201
UseEPC_G1C1HF ... 201
UseISO14443A ... 201

© 2011 MicroEM

UseISO14443B ... 201
Методы ... 201
Execute ... 201
WriteData ... 202
Свойства ... 202
UsedReaders ... 202
ExpectedCardsStandards ... 202
UseICode1 ... 202
UseISO15693 ... 202
UseEPC_G1C1HF ... 203
UseISO14443A ... 203
UseISO14443B ... 203
AllDataAsTextString ... 203
AllDataAsHexString ... 203
Методы ... 203
Execute ... 203
AutoSearch ... 204
Свойства ... 204
ExpectedCardsCount ... 204
UsedReaders ... 204
ExpectedCardsStandards ... 204
UseICode1 ... 204
UseISO15693 ... 205
UseEPC_G1C1HF ... 205
UseISO14443A ... 205
UseISO14443B ... 205
SessionState ... 205
CurrentTagsList ... 205
AutoSwitchRF ... 205
Методы ... 206
StartSession ... 206
PauseSession ... 206
ResetSession ... 206
Типы ... 206
XID_eSRC_AS_SessionState ... 206
3 Компонента 1С ... 207
Введение ... 207
Общие требования ... 207
Описание работы с драйвером ... 207
Начало работы с драйвером ... 207
Чтение данных из меток ... 208
Запись данных в метки ... 209
Состояния переходов между режимами ... 209
Завершение работы ... 209
Внешние события от драйвера ... 209
Свойства ... 0
Методы ... 210
ПолучитьОшибку ... 210
Подключить ... 210
МассивЗначений ... 210
МодельУстройства ... 210
Порт ... 210
АдресНаШине ... 211
АдресIP ... 211

ИнтерфейсСвязи ... 211
НакапливатьМетки ... 211
РазделительМетки ... 211
ПредельноеКолвоМеток ... 212
Таймаут ... 212
МаксКоличМеток ... 212
Отключить ... 212
ВключитьАктивныйРежим ... 212
ВыключитьАктивныйРежим ... 213
ОткрытьСессию ... 213
ЗакрытьСессию ... 213
СессияОткрыта ... 214
НайтиМетки ... 214
СчитатьСледДанные ... 214
ЗаписатьДанные ... 215
СобытиеОбработано ... 215
ПодатьСигнал ... 215
ПоддерживаетПБ ... 216
УстановитьПБ ... 216
ПолучитьПБ ... 217
ТестУстройства ... 217
МассивЗначений ... 217
МодельУстройства ... 217
Порт ... 218
АдресНаШине ... 218
АдресIP ... 218
ИнтерфейсСвязи ... 218
НакапливатьМетки ... 218
РазделительМетки ... 219
ПредельноеКолвоМеток ... 219
Таймаут ... 219
МаксКоличМеток ... 219
События ... 219
ОшибкаВДрайвере ... 219
МеткаВПоле ... 219
НетМетокВПоле ... 220
ЧтениеМеток ... 220

Глава VI Порядок взаимодействия объектов библиотеки при решении различных задач 222

1 Общие принципы ... 222
2 Работа с базовыми объектами ... 222
3 Работа с упрощенными объектами ... 223

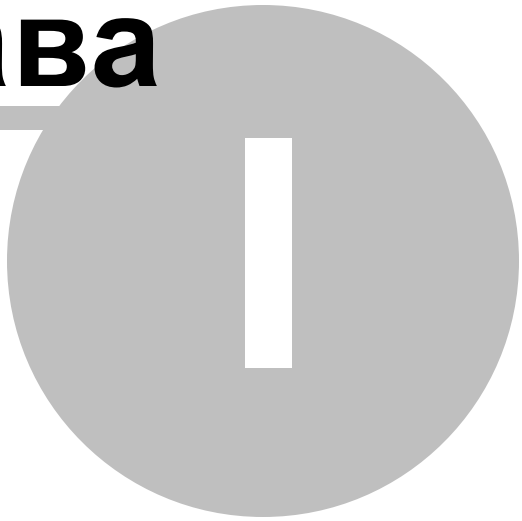
Глава VII Примеры использования библиотеки 226

1 Примеры работы с базовыми объектами ... 226
Пример работы с метками стандарта ISO15693 ... 226
2 Примеры работы с упрощенными объектами ... 228
Пример работы с объектом XID_SimpleReadersControl ... 228

© 2011 MicroEM

Index **231**

Глава



1 Введение

1.1 О библиотеке

Библиотека **XID_Designer** предназначена для упрощения построения высокоуровневых приложений систем идентификации (**RFID**).

Библиотека предоставляет разработчику набор объектов для работы с устройствами идентификации и обеспечивает поддержку следующих стандартов бесконтактных карт (меток):

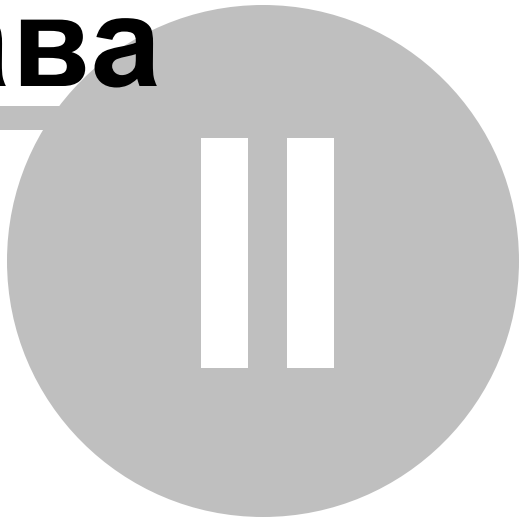
- ISO15693
- ISO14443A
- ISO14443B
- EPC Class1 Gen2

Кроме того, объекты библиотеки предоставляют интерфейсы, упрощающие работу с такими типами карт, как:

- Philips ICodeSLI
- Philips Mifare

© 2011 MicroEM

Глава



2 Установка библиотеки на систему

2.1 Установка пакета разработчика

Пакет разработчика **XID_DesignerLib.exe** может быть установлен на следующие операционные системы:

- ✓ MS Windows 98
- ✓ MS Windows 2000
- ✓ MS Windows XP

При установке пакета происходит копирование следующих необходимых для работы библиотеки файлов в каталог "*Program Files\MicroEM\XID_Designer Library*":

- XID_Designer.dll (основной файл, регистрируется в системе)
- ComEnume.dll
- clscrfl.dll
- RH_Control.dll
- msjava.dll
- FeIsc.dll
- FeCom.dll
- FeUsb.dll
- fetcp.dll
- msvcr80.dll
- msvcp80.dll
- msvcm80.dll
- Microsoft.VC80.CRT.manifest

В папке "*Program Files\MicroEM\Demo*" будут размещены следующие проекты с демонстрацией применения библиотеки:

- "MSVS 6.0 Visual Basic" - Visual Basic (проект MS Visual Studio 6.0)
- "MSVS 2005 Visual Basic" - Visual Basic (проект MS Visual Studio 2005)
- "MSVS 2005 Visual C++" - Visual C++ (проект MS Visual Studio 2005)
- "MSVS 2005 Visual C++ CLR" - Visual C++ Common Language Runtime (проект MS Visual Studio 2005)
- "MSVS 2005 Visual J#" - Visual J# (проект MS Visual Studio 2005)
- "MSVS 2005 Visual C#" - Visual C# (проект MS Visual Studio 2005)
- "Borland Delphi 7.0" - проект Delphi 7.0
- "Borland C++ Builder 5.0" - проект C++ Builder 5.0
- "Html" - страницы Html с применением javascript и vbscript
- "1С Предприятие 8.0" - модули 1С Предприятие 8.0

После завершения установки библиотека готова к использованию.

2.2 Интеграция библиотеки в пакет установки приложения

Для интеграции библиотеки при формировании пакета установки приложения, пользователю следует включить следующие файлы в пакет:

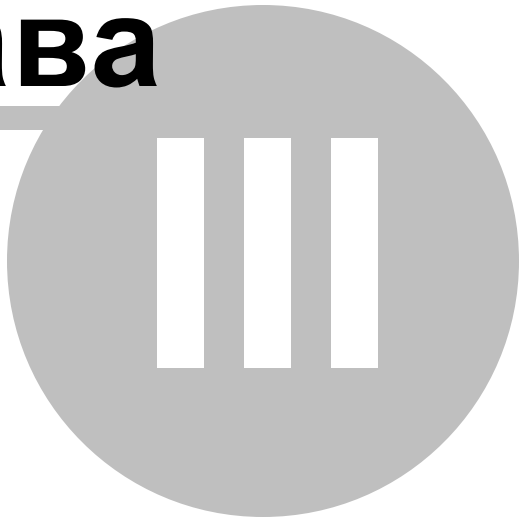
- XID_Designer.dll (основной файл библиотеки, следует зарегистрировать в системе - посредством "regsvr32.exe")
- ComEnume.dll
- clscrfl.dll

- RH_Control.dll
- msjava.dll
- FeIsc.dll
- FeCom.dll
- FeUsb.dll
- fetcp.dll
- папка Microsoft.VC80.CRT

Файлы, кроме основного, должны находиться в одном каталоге с основным, либо частично/полностью в системном каталоге Windows.

© 2011 MicroEM

Глава



3 Подключение библиотеки к наиболее распространенным типам проектов

3.1 MS Visual Studio 6.0

3.1.1 Visual C++ 6.0

Для подключения библиотеки к данному проекту следует добавить в заголовочный файл (например, "stdafx.h") следующую строку:

```
#import "libid:40B7CB45-E181-4019-B57F-BCB714390A1C" no_namespace
```

или

```
#import "<путь к файлу>\XID_Designer.dll" no_namespace
```

3.1.2 Visual Basic 6.0

Для подключения библиотеки к данному проекту следует выполнить следующие действия:

1) перейти в меню "Project->References...";
2) отметить флажком компонент "XID_Designer 1.0 Type Library";
3) нажать "OK".

3.2 MS Visual Studio 2003

3.2.1 Visual C++ 2003

Для подключения библиотеки к данному проекту следует добавить в заголовочный файл (например, "stdafx.h") следующую строку:

```
#import "libid:40B7CB45-E181-4019-B57F-BCB714390A1C" no_namespace
```

или

```
#import "<путь к файлу>\XID_Designer.dll" no_namespace
```

3.2.2 Visual Basic 2003

Для подключения библиотеки к данному проекту следует выполнить следующие действия:

1) нажать правой кнопкой мыши на папку "References" в дереве "Solution Explorer-><имя проекта>" или на само <имя проекта>;
2) в появившемся контекстном меню выбрать пункт "Add Reference...";

либо:

1) выбрать из меню "Project->AddReference...";

далее:

3) перейти на закладку "COM";
4) выбрать компонент "XID_Designer 1.0 Type Library";
5) нажать "Select";

© 2011 MicroEM

6) нажать "OK".

3.2.3 Visual C# 2003

Для подключения библиотеки к данному проекту следует выполнить следующие действия:
1) нажать правой кнопкой мыши на папку "References" в дереве "Solution Explorer-><имя проекта>" или на само <имя проекта>;
2) в появившемся контекстном меню выбрать пункт "Add Reference...";
либо:
1) выбрать из меню "Project->AddReference...";
далее:
3) перейти на закладку "COM";
4) выбрать компонент "XID_Designer 1.0 Type Library";
5) нажать "Select";
6) нажать "OK".

3.2.4 Visual J# 2003

Для подключения библиотеки к данному проекту следует выполнить следующие действия:
1) нажать правой кнопкой мыши на папку "References" в дереве "Solution Explorer-><имя проекта>" или на само <имя проекта>;
2) в появившемся контекстном меню выбрать пункт "Add Reference...";
либо:
1) выбрать из меню "Project->AddReference...";
далее:
3) перейти на закладку "COM";
4) выбрать компонент "XID_Designer 1.0 Type Library";
5) нажать "Select";
6) нажать "OK".

3.3 MS Visual Studio 2005/2008

3.3.1 Visual C++ 2005/2008

Для подключения библиотеки к данному проекту следует добавить в заголовочный файл (например, "stdafx.h") следующую строку:

```
#import "libid:40B7CB45-E181-4019-B57F-BCB714390A1C" no_namespace
```

3.3.2 Visual Basic 2005/2008

Для подключения библиотеки к данному проекту следует выполнить следующие действия:
1) нажать правой кнопкой мыши на <имя проекта>;
2) в появившемся контекстном меню выбрать пункт "Add Reference...";
либо:
1) выбрать из меню "Project->AddReference...";

либо:
открыть свойства проекта "Project->Properties...";
1) перейти на закладку "References";
2) нажать кнопку "Add...";
далее:
3) перейти на закладку "COM";
4) выбрать компонент "XID_Designer 1.0 Type Library";
5) нажать "OK".

3.3.3 Visual C# 2005/2008

Для подключения библиотеки к данному проекту следует выполнить следующие действия:
1) нажать правой кнопкой мыши на папку "References" в дереве "Solution Explorer-><имя проекта>" или на само <имя проекта>;
2) в появившемся контекстном меню выбрать пункт "Add Reference...";
либо:
1) выбрать из меню "Project->AddReference...";
далее:
3) перейти на закладку "COM";
4) выбрать компонент "XID_Designer 1.0 Type Library";
5) нажать "OK".

3.3.4 Visual J# 2005/2008

Для подключения библиотеки к данному проекту следует выполнить следующие действия:
1) нажать правой кнопкой мыши на папку "References" в дереве "Solution Explorer-><имя проекта>" или на само <имя проекта>;
2) в появившемся контекстном меню выбрать пункт "Add Reference...";
либо:
1) выбрать из меню "Project->AddReference...";
далее:
3) перейти на закладку "COM";
4) выбрать компонент "XID_Designer 1.0 Type Library";
5) нажать "OK".

3.3.5 Visual C++ CLR 2005/2008

Для подключения библиотеки к данному проекту следует выполнить следующие действия:
1) нажать правой кнопкой мыши на <имя проекта>;
2) в появившемся контекстном меню выбрать пункт "Add Reference...";
либо:
1) выбрать из меню "Project->AddReference...";
либо:
открыть свойства проекта "Project->Properties...";
1) перейти на закладку "References";
2) нажать кнопку "Add...";
далее:
3) перейти на закладку "COM";
4) выбрать компонент "XID_Designer 1.0 Type Library";

© 2011 MicroEM

5) нажать "OK".

3.4 Borland Delphi 7.0

Для подключения библиотеки к данному проекту следует выполнить следующие действия:

1) выбрать меню "Project->Import Type Library...";
2) выбрать библиотеку "XID_Designer 1.0 Type Library";
3) выбрать страницу палеты ("Palette page"), куда будут добавлены значки объектов библиотеки (например, "ActiveX");
4) нажать "Install...".

После выполнения указанных операций библиотека будет интегрирована во внутреннюю библиотеку Delphi, а новые компоненты будут доступны на выбранной панели ("Palette page").

Перед открытием демо-проекта на Delphi требуется импортировать библиотеку (см. выше), а также добавить элементы ActiveX на панель инструментов (берутся из той же библиотеки).

3.5 Borland C++ Builder 5.0

Для подключения библиотеки к данному проекту следует выполнить следующие действия:

1) выбрать меню "Project->Import Type Library...";
2) выбрать библиотеку "XID_Designer 1.0 Type Library";
3) выбрать страницу палеты ("Palette page"), куда будут добавлены значки объектов библиотеки (например, "ActiveX");
4) нажать "Install...".

После выполнения указанных операций библиотека будет интегрирована во внутреннюю библиотеку Delphi, а новые компоненты будут доступны на выбранной панели ("Palette page").

© 2011 MicroEM

Глава

4 Порядок работы с объектами COM в наиболее распространенных типах проектов

4.1 Создание объектов

MS Visual C++

IXID_ManagerPtr mgr;

- создает указатель на интерфейс mgr типа IXID_Manager из библиотеки XID_DesignerLib посредством создания объекта-оболочки в виде умного указателя IXID_ManagerPtr.

mgr.CreateInstance("XID_DesignerLib.XID_Manager");

- создает объект типа XID_Manager и записывает указатель на его основной интерфейс в переменную mgr.
При этом счетчик ссылок на объект становится равным 1.

MS Visual C++ CLR

XID_DesignerLib::XID_Manager ^ mgr;

- создает переменную mgr типа XID_Manager из библиотеки XID_DesignerLib.

mgr = gcnew XID_DesignerLib::XID_Manager;

- создает объект типа XID_Manager из библиотеки XID_DesignerLib и присваивает его переменной mgr.
При этом счетчик ссылок на объект становится равным 1.

MS Visual Basic

Dim mgr As XID_DesignerLib.XID_Manager

- создает переменную mgr типа XID_Manager из библиотеки XID_DesignerLib.

mgr = New XID_DesignerLib.XID_Manager

- создает объект типа XID_Manager из библиотеки XID_DesignerLib и присваивает его переменной mgr.
При этом счетчик ссылок на объект становится равным 1.

MS Visual C#

XID_DesignerLib.XID_Manager mgr;

- создает переменную mgr типа XID_Manager из библиотеки XID_DesignerLib.

mgr = new XID_DesignerLib.XID_Manager();

© 2011 MicroEM

- создает объект типа XID_Manager из библиотеки XID_DesignerLib и присваивает его переменной mgr.
При этом счетчик ссылок на объект становится равным 1.

MS Visual J#

XID_DesignerLib.XID_Manager mgr;

- создает переменную mgr типа XID_Manager из библиотеки XID_DesignerLib.

mgr = new XID_DesignerLib.XID_ManagerClass();

- создает объект типа XID_Manager из библиотеки XID_DesignerLib и присваивает его переменной mgr.
При этом счетчик ссылок на объект становится равным 1.

Borland Delphi

Способ 1

Помещаем нужный объект из панели на форму

Способ 2

var mgr: TXID_Manager;

- создает переменную mgr типа TXID_Manager из библиотеки XID_DesignerLib.

mgr := TXID_Manager.Create(self);

- создает объект типа TXID_Manager из библиотеки XID_DesignerLib и присваивает его переменной mgr.
При этом счетчик ссылок на объект становится равным 1.

Borland Builder

Способ 1

Помещаем нужный объект из панели на форму

Способ 2

XID_ManagerPtr mgr;

- создает умный указатель mgr типа XID_ManagerPtr из библиотеки XID_DesignerLib.

mgr.CreateInstance(OLESTR("XID_DesignerLib.XID_Manager"));

```
- создает объект типа XID_Manager из библиотеки XID_DesignerLib и присваивает его переменной mgr.
При этом счетчик ссылок на объект становится равным 1.
```

Html

```
Способ 1

Создаем глобальный объект:
<object id="mgr" classid="clsid:7EC51798-270C-40C9-8AFA-7E8EBD8D2AC1" codebase="XID_DesignerLib.cab">
</object>

Способ 2 (javascript)

var mgr = new ActiveXObject('XID_DesignerLib.XID_Manager');

- создает объект типа XID_Manager из библиотеки XID_DesignerLib и присваивает его переменной mgr.
При этом счетчик ссылок на объект становится равным 1.
```

1С Предприятие 8.0

```
mgr = ПолучитьCOMОбъект("", "XID_DesignerLib.XID_Manager");

- создает объект типа XID_Manager из библиотеки XID_DesignerLib и присваивает его переменной mgr.
При этом счетчик ссылок на объект становится равным 1.
```

4.2 Доступ к методам объектов

MS Visual C++

```
mgr->DetectDevices();
```

MS Visual C++ CLR

```
mgr->DetectDevices();
```

MS Visual Basic

```
mgr.DetectDevices()
```

MS Visual C#

```
mgr.DetectDevices();
```

MS Visual J#

```
mgr.DetectDevices();
```

Borland Delphi

```
mgr.DetectDevices;
```

Borland Builder

```
mgr->DetectDevices();
```

Html (vbscript)

```
mgr.DetectDevices

Внимание!!! Поскольку в языках-скриптах доступны только интерфейсы типа IDispatch,
все свойства и методы интерфейса типа IUnknown доступны также из родительского
интерфейса типа IDispatch, но имеют  атрибут скрытых, чтобы "не мешаться" в тех
средах, где IUnknown доступен. Например, метод XID_ControlX.ComplexFuncs.
FullAnticollision.do_FullAnticollision() также расположен в родительском интерфейсе:
XID_ControlX.ComplexFuncs.do_FullAnticollision().
```

1С Предприятие 8.0

```
mgr.DetectDevices();
```

4.3 Доступ к свойствам объектов

MS Visual C++

```
dev = devs->Item[i];
dev->ID = Value;
Value = dev->ID;
```

MS Visual C++ CLR

```
dev = devs[i]
dev->ID = Value
Value = dev->ID
```

MS Visual Basic

```
dev = devs(i)
dev.ID = Value
Value = dev.ID
```

© 2011 MicroEM

MS Visual C#

```
dev = devs[i];
dev.ID = Value;
Value = dev.ID;
```

MS Visual J#

```
dev = devs.get_Item(i);
dev.set_ID(Value);
Value = dev.get_ID();
```

Borland Delphi

```
dev := devs[i];
dev.ID := Value;
Value := dev.ID;
```

Borland Builder

```
dev = devs[i];
dev->ID = Value;
Value = dev->ID;
```

Html (vbscript)

```
dev = devs(i)
dev.ID = Value
Value = dev.ID
```

Внимание!!! Поскольку в языках-скриптах доступны только интерфейсы типа IDispatch, все свойства и методы интерфейса типа IUnknown доступны также из родительского интерфейса типа IDispatch, но имеют атрибут скрытых, чтобы "не мешаться" в тех средах, где IUnknown доступен. Например, свойство XID_ControlX.ComplexFuncs.FullAnticollision.Readers также расположено в родительском интерфейсе: XID_ControlX.ComplexFuncs.Readers.

1С Предприятие 8.0

```
dev = devs(i);
dev.ID = Value;
Value = dev.ID;
```

4.4 Доступ к интерфейсам объектов

MS Visual C++

```
cx->ComplexFuncs->[…]
```

MS Visual C++ CLR

```
cx->ComplexFuncs->[…]
```

MS Visual Basic

```
cx.ComplexFuncs.[…]
```

MS Visual C#

```
cx.ComplexFuncs.[…]
```

MS Visual J#

```
cx.get_ComplexFuncs().[…]
```

Borland Delphi

```
cx.ComplexFuncs.[…]
```

Borland Builder

```
cx->ComplexFuncs->[…]
```

Html (vbscript)

```
cx.ComplexFuncs.[…]
```

1С Предприятие 8.0

```
cx.ComplexFuncs.[…]
```

4.5 Удаление объектов

MS Visual C++

```
mgr.Release();

- понижает счетчик ссылок на объект mgr на 1. При достижении значения счетчика, равного 0, объект автоматически удаляется из памяти.
```

MS Visual C++ CLR

При глобальном определении переменной, объект удаляется по завершении работы приложения.
При локальном определении - объект удаляется при выходе из функции.

MS Visual Basic

При глобальном определении переменной, объект удаляется по завершении работы приложения.
При локальном определении - объект удаляется при выходе из функции.

MS Visual C#

При глобальном определении переменной, объект удаляется по завершении работы приложения.
При локальном определении - объект удаляется при выходе из функции.

MS Visual J#

При глобальном определении переменной, объект удаляется по завершении работы приложения.
При локальном определении - объект удаляется при выходе из функции.

Borland Delphi

Способ 1
Помещаем нужный объект из панели на форму

В этом случае объект удалится при удалении формы.

Способ 2

mgr.Destroy;

- удаляет объект mgr.

Borland Builder

При глобальном определении переменной, объект удаляется по завершении работы приложения.
При локальном определении - объект удаляется при выходе из функции.

Html

При глобальном определении переменной, объект удаляется по завершении работы приложения.
При локальном определении - объект удаляется при выходе из функции.

1С Предприятие 8.0

При глобальном определении переменной, объект удаляется по завершении работы приложения.
При локальном определении - объект удаляется при выходе из функции.

4.6 Реализация событий объектов

MS Visual C++

В некотором классе среди родителей пишем:

```
public IDispEventImpl<1, CMyClass, &__uuidof
( _IXID_SimpleReadersControlEvents), &LIBID_XID_DesignerLib, /
*wMajor =*/ 1, /*wMinor =*/ 0>
```

где CMyClass - имя наследующего класса;

в теле описания класса:

```
void __stdcall OnTagEvent( IXID_TagsList * piTagsList,
VARIANT_BOOL IsInserted);
BEGIN_SINK_MAP(CMyClass)
    SINK_ENTRY_EX(1, __uuidof( _IXID_SimpleReadersControlEvents),
1, OnTagEvent)
END_SINK_MAP()

IXID_SimpleReadersControlPtr m_SRC;
CMyClass()
{
    m_SRC.CreateInstance("XID_DesignerLib.
XID_SimpleReadersControl");
}

~CMyClass()
{
    m_SRC.Release();
}
```

в теле класса:

```
void __stdcall CMyClass::OnTagEvent( IXID_TagsList * piTagsList,
VARIANT_BOOL IsInserted)
{
}
```

при создании экземпляра класса CMyClass:

```
CMyClass* pMC = new CMyClass();
IUnknown* piUnk = NULL;
pMC->m_SRC.QueryInterface(IID_IUnknown, (void**)&piUnk);
pMC->DispEventAdvise(piUnk);
piUnk->Release();
```

перед удалением экземпляра класса CMyClass:

```
pMC->m_SRC->AutoSearch->ResetSession(TRUE);
IUnknown* piUnk = NULL;
```

© 2011 MicroEM

```
pMC->m_SRC.QueryInterface(IID_IUnknown, (void**)&piUnk);
pMC->DispEventUnadvise(piUnk);
piUnk->Release();
delete pMC;
```

MS Visual C++ CLR

В теле класса формы:

```
void src_OnSessionPaused(void)
{
    MessageBox::Show("src_OnSessionPaused!!!");
}
```

После создания объекта:

```
    src->OnSessionPaused += gcnew XID_DesignerLib::
_IXID_SimpleReadersControlEvents_OnSessionPausedEventHandler(this, &test_clr::Form1::
src_OnSessionPaused);
```

MS Visual Basic

При определении типа объекта указывать ключевое слово "WithEvents":

```
Public WithEvents src As New XID_SimpleReadersControl
```

далее:

для Visual Studio 2005:

```
Private Sub src_OnSessionPaused() Handles src.OnSessionPaused
    MsgBox("OnSessionPaused")
End Sub
```

для Visual Studio 6.0:

```
Private Sub src_OnSessionPaused()
    MsgBox("OnSessionPaused")
End Sub
```

MS Visual C#

В теле класса формы:

```
public void src_OnSessionPaused()
{
    MessageBox.Show("src_OnSessionPaused!!!");
}
```

После создания объекта:

```
    src.OnSessionPaused += new XID_DesignerLib.
_IXID_SimpleReadersControlEvents_OnSessionPausedEventHandler(src_OnSessionPaused);
```

MS Visual J#

В теле класса формы:

```
        public void src_OnSessionPaused()
        {
            MessageBox.Show("src_OnSessionPaused!!!");
        }
```

После создания объекта:

```
            src.add_OnSessionPaused(new XID_DesignerLib.
_IXID_SimpleReadersControlEvents_OnSessionPausedEventHandler(src_OnSessionPaused));
```

Borland Delphi

Обработчик события создается автоматически средствами Delphi:

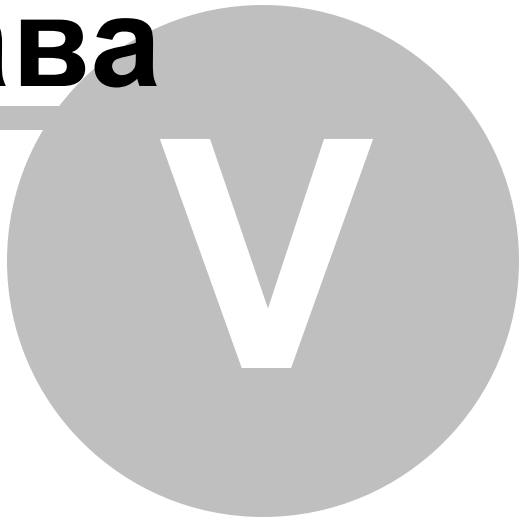
```
procedure TForm1.srcSessionPaused(Sender: TObject);
begin
  MessageBox(self.Handle, 'srcSessionPaused', '123', MB_OK);
end;
```

Borland Builder

Html (vbscript)

1С Предприятие 8.0

Глава



5 Описание типов объектов библиотеки

Ниже описаны типы объектов данной библиотеки.

Примечание Значком * помечены разделы и объекты, находящиеся на стадии разработки.

5.1 Базовые объекты

Данная группа объединяет фундаментальные компоненты библиотеки, необходимые для построения систем идентификации.

5.1.1 XID_Manager

XID_Manager- это один из основных объектов библиотеки наряду с **XID_SimpleReadersControl**

Основные функции данного объекта в текущей версии библиотеки - поиск подключенных устройств идентификации и составление списка найденных устройств.

5.1.1.1 Методы

5.1.1.1.1 DetectDevices

```
DetectDevices()
```

Производит поиск поддерживаемых устройств всех поддерживаемых производителей в системе (может длиться довольно долго, предпочтительнее использовать **DetectVendorDevices**).

5.1.1.1.2 DetectVendorDevices

```
DetectVendorDevices
(
    [in] string Vendor
)
```

Производит поиск поддерживаемых устройств от конкретного производителя **Vendor** в системе (такой поиск предпочтительнее, так как он занимает намного меньше времени, нежели использование **DetectDevices**).

5.1.1.1.3 GetDevices

```
GetDevices
(
    XID_DevicesList devs
)
```

Очищает список устройств **devs** и записывает в него найденные устройства.

devs должен быть создан вызывающим приложением заранее.

5.1.2 XID_DevicesList

XID_DevicesList является одним из базовых объектов библиотеки и представляет собой список устройств с набором свойств и методов, необходимых для добавления, удаления новых элементов и пр..

5.1.2.1 Методы

5.1.2.1.1 Clear

Clear()

Очищает список устройств.

5.1.2.1.2 Add

Add
(
    XID_Device dev
)

Добавляет устройство **dev** в список.

Объект **dev** должен быть предварительно создан.

5.1.2.1.3 AddDevicesList

AddDevicesList
(
    XID_DevicesList devs
)

Добавляет список устройств **devs** в список.

Объект **devs** должен быть предварительно создан.

5.1.2.1.4 Remove

```
Remove
(
    short Index
)
```

Удаляет объект устройства с индексом **Index** из внутреннего массива списка устройств.

5.1.2.1.5 GetDevicesByClass

```
GetDevicesByClass
(
    XID_DevicesList devs,
    XID_eDeviceClass devclass
)
```

Записывает из текущего списка устройств в **devs** устройства, относящиеся к классу **devclass**.

Объект **devs** должен быть предварительно создан.

5.1.2.1.6 GetReaders

```
GetReaders
(
    XID_DevicesList devs
)
```

Записывает из текущего списка устройств в **devs** устройства, относящиеся к классу XID_eDeviceClass.**xidDevice_Reader**

Объект **devs** должен быть предварительно создан.

5.1.2.1.7 CopyTo

```
CopyTo
(
    XID_DevicesList devs
)
```

Копирует содержимое текущего списка устройств в **devs**.

Объект **devs** должен быть предварительно создан.

© 2011 MicroEM

5.1.2.2 Свойства

5.1.2.2.1 Count

[out] short Count

Возвращает текущее количество устройств в списке.

5.1.2.2.2 Item

[out] XID_Device Item
(
short Index
)

Возвращает указатель на объект устройства с индексом **Index** из внутреннего массива списка устройств (счетчик ссылок на объект данного устройства при этом увеличивается на 1). Это свойство также доступно при указании индекса устройства в скобках сразу же после имени переменной-держателя списка устройств (см. Доступ к свойствам объектов).

5.1.2.2.3 GetDeviceIndex

[out] short GetDeviceIndex
(
XID_Device dev
)

Возвращает индекс устройства из списка устройств, идентичного устройству **dev**.

Объект **dev** должен быть предварительно создан.

Если такое устройство отсутствует в списке, возвращается "-1".

5.1.3 XID_Device

XID_Device является одним из базовых объектов библиотеки и включает в себя набор общих свойств устройства системы идентификации (в большинстве случаев это устройство представляет собой считыватель).

5.1.3.1 Методы

5.1.3.1.1 CloneFrom

CloneFrom
(
XID_Device dev

)

Устанавливает в текущем устройстве свойства, идентичные свойствам **dev**.

Объект **dev** должен быть предварительно создан.

5.1.3.2 Свойства

5.1.3.2.1 ComPort

[in, out] byte ComPort

Коммуникационный порт устройства.

5.1.3.2.2 ID

[in, out] string ID

Идентификатор устройства в шестнадцатеричном виде.

5.1.3.2.3 BSA

[in, out] byte BSA

Адрес шины устройства (для устройств на шине RS485).

5.1.3.2.4 Model

[in, out] string Model

Модель устройства в виде строки.

5.1.3.2.5 Version

[in, out] string Version

Версия устройства в шестнадцатеричном виде.

5.1.3.2.6 Vendor

[in, out] string Vendor

© 2011 MicroEM

Наименование производителя устройства в виде строки.

5.1.3.2.7 CommBaudRate

[in, out] XID_eCommBaudRate CommBaudRate

Скорость обмена с устройством по порту.

5.1.3.2.8 CommunicationType

[in, out] XID_eCommunicationType CommunicationType

Тип интерфейса связи с устройством.

5.1.3.2.9 DeviceClass

[in, out] XID_eDeviceClass DeviceClass

Класс устройства.

5.1.3.2.10 IsEqualTo

[out] boolean IsEqualTo
(
XID_Device dev
)

Возвращает истину, если текущее устройство тождественно устройству **dev**.

Объект **dev** должен быть предварительно создан.

5.1.3.3 Типы

5.1.3.3.1 XID_eDeviceClass

enum XID_eDeviceClass

Данный перечислимый тип объединяет типичные классы поддерживаемых библиотекой устройств RFID:

xidDevice_Unknown	идентифицирует любое неизвестное устройство
xidDevice_Reader	идентифицирует устройство класса "считыватель"

5.1.3.3.2 XID_eCommBaudRate

enum XID_eCommBaudRate

Данный перечислимый тип объединяет скорости, на которых может выполняться передача для RS-интерфейсов (бод):

xidCBR_Default	скорость по умолчанию (для разных устройств - своя)
xidCBR_2400	
xidCBR_4800	
xidCBR_9600	
xidCBR_14400	
xidCBR_19200	
xidCBR_28800	
xidCBR_38400	
xidCBR_57600	
xidCBR_76800	
xidCBR_115200	

5.1.3.3.3 XID_eCommunicationType

enum XID_eCommunicationType

Данный перечислимый тип объединяет типы интерфейсов связи устройства с ПК:

xidCommType_RS232	RS232
xidCommType_RS485	RS485
xidCommType_VirtCOM	виртуальный COM-порт
xidCommType_USB	USB-устройство
xidCommType_LAN	устройство на шине Ethernet

5.1.4 XID_ControlX

XID_ControlX является базовым объектом управления устройствами системы идентификации и предоставляет пользователю возможность работы устройствами различных типов и метками различных стандартов посредством выполнения команд, размещенных в древовидной структуре, объединяющей соответствующие свойства и методы.

5.1.4.1 Методы

5.1.4.1.1 SelectDevice

SelectDevice
(

```
        XID_Device dev
)
```

Выбирает для текущего объекта **XID_ControlX** устройство **dev**, с которым дальше будет производиться работа.

5.1.4.1.2 LinkDevice

```
LinkDevice()
```

Производит подключение выбранного методом **SelectDevice** устройства к текущему объекту **XID_ControlX**.

5.1.4.1.3 UnlinkDevice

```
UnlinkDevice()
```

Производит отключение выбранного методом **SelectDevice** устройства от текущего объекта **XID_ControlX**.

5.1.4.1.4 LinkDevices

```
LinkDevices
(
        XID_DevicesList devs
)
```

Производит подключение списка устройств **devs** к текущему объекту **XID_ControlX**.

5.1.4.1.5 UnlinkAllDevices

```
UnlinkAllDevices()
```

Производит отключение всех устройств, подключенных к текущему объекту **XID_ControlX**.

5.1.4.2 Свойства

5.1.4.2.1 LastError

```
[out] XID_eGeneralErrorCodes LastError
```

Код ошибки при выполнении последней команды хоста (исполняющего компонента библиотеки).

5.1.4.2.2 LastErrorDescription

[out] string LastErrorDescription

Возвращает описание последней ошибки в виде символьной строки.

5.1.4.2.3 Language

[in, out] XID_eLanguage Language

Текущий язык для выдачи сообщений библиотеки, а также описания ошибки.

5.1.4.2.4 TimeOut

[in, out] long TimeOut

Время ожидания ответа от устройства в миллисекундах.

5.1.4.3 Типы

5.1.4.3.1 XID_eSlotErrorCodes

enum XID_eSlotErrorCodes

Данный перечислимый тип объединяет коды ошибок для временных интервалов ответов карт:

xidSlotErr_OK	Нет ошибок
xidSlotErr_IsEmpty	Слот пустой (нет карты)
xidSlotErr_WrongCRC	Ошибка при проверке CRC
xidSlotErr_Collision	Коллизия в слоте
xidSlotErr_TimeOut	Превышено время ожидания ответа
xidSlotErr_UndefinedError	Неизвестная ошибка

5.1.4.3.2 XID_eGeneralErrorCodes

enum XID_eGeneralErrorCodes

Данный перечислимый тип объединяет общие коды ошибок при работе с устройствами:

xidErr_OK	Нет ошибок
xidErr_NoCard	Нет карты
xidErr_WrongKey	Неверный ключ

© 2011 MicroEM

xidErr_WrongMode	Неверный режим
xidErr_ExecutionFailed	Ошибка выполнения команды
xidErr_TimeOut	Вышло время ожидания ответа на команду
xidErr_WrongCRC	Неверный CRC
xidErr_LinkFailed	Ошибка при подключении к устройству
xidErr_UpdateFailed	Ошибка обновления прошивки
xidErr_InvalidParameter	Неверный параметр
xidErr_CommandNotSupported	Команда не поддерживается
xidErr_HostIsBusy	Хост (исполняющий компонент библиотеки) занят
xidErr_TooManyReaders	Слишком много считывателей выбрано
xidErr_CommandIsInProgress	Команда находится в процессе выполнения
xidErr_InvalidValue	Неверное значение
xidErr_UndefinedError	Неопределённая ошибка

5.1.4.3.3 XID_eCodedTimeSlots

enum XID_eCodedTimeSlots

Данный перечислимый тип объединяет возможные значения количества временных интервалов, выделяемых для ответа метки:

xidTimeSlots_1	1 временной интервал
xidTimeSlots_4	4 временных интервала
xidTimeSlots_8	8 временных интервалов
xidTimeSlots_16	16 временных интервалов
xidTimeSlots_32	32 временных интервала
xidTimeSlots_64	64 временных интервала
xidTimeSlots_128	128 временных интервалов
xidTimeSlots_256	256 временных интервалов
xidTimeSlots_512	512 временных интервалов

5.1.4.3.4 XID_eLanguage

enum XID_eLanguage

Данный перечислимый тип объединяет поддерживаемые библиотекой языки (на этих языках может выдаваться текстовое сообщение об ошибке):

xidLanguage_English	Английский
xidLanguage_Russian	Русский

© 2011 MicroEM

5.1.4.4 События

5.1.4.4.1 OnActivateISO14443A

```
OnActivateISO14443A
(
    string UID
)
```

Данное событие возникает при обнаружении карты стандарта ISO14443A (Mifare) в поле одного из выбранных считывателей при выполнении команды .ControlX.ComplexFuncs.FullAnticollision, либо при открытой сессии группы команд **AutoSearch**.

UID - строка-идентификатор обнаруженной в поле метки (в фотмате HEX).

5.1.4.4.2 OnActivateISO14443B

```
OnActivateISO14443B
(
    string UID
)
```

Данное событие возникает при обнаружении карты стандарта ISO14443B (Mifare) в поле одного из выбранных считывателей при выполнении команды .ControlX.ComplexFuncs.FullAnticollision, либо при открытой сессии группы команд **AutoSearch**.

UID - строка-идентификатор обнаруженной в поле метки (в фотмате HEX).

5.1.4.4.3 OnActivateISO15693

```
OnActivateISO15693
(
    string UID
)
```

Данное событие возникает при обнаружении карты стандарта ISO15693 (ICode) в поле одного из выбранных считывателей при выполнении команды .ControlX.ComplexFuncs.FullAnticollision, либо при открытой сессии группы команд **AutoSearch**.

UID - строка-идентификатор обнаруженной в поле метки (в фотмате HEX).

5.1.4.5 Сложные функции (Complex Functions)

```
IXID_ComplexFuncs ComplexFuncs
```

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.ComplexFuncs**.

Предоставляемые данным интерфейсом функции предназначены для комплексной работы с метками различных стандартов.

5.1.4.5.1 Full Anticollision

IXID_Cmd_Complex_FullAnticollision FullAnticollision

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.ComplexFuncs.FullAnticollision**.

Функция **FullAnticollision** предназначена для автоматизированного поиска карт различных стандартов при помощи указанных считывателей.

Включение и выключение поля считывателя функция выполняет автоматически.

5.1.4.5.1.1 Свойства

[in, out] XID_DevicesList Readers

Список считывателей, которые должны будут участвовать в работе с метками.

Перед выполнением функции, **Readers** должны быть подключены к текущему объекту **XID_ControlX** посредством вызова функции **LinkDevices**.

[in, out] XID_TagsList tags

Список меток, из которого будут браться исходные данные и куда будут помещены результаты выполнения.

[out] IXID_UseStandards in_Standards

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно установить стандарты меток, над которыми следует выполнять операции.

Свойства **in_Standards** описаны в подразделах данного раздела.

[in, out] boolean UseEPC_G1C1HF

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **EPC, Gen1, Class1, 13.56MHz**.

[in, out] boolean UseISO14443A

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO14443A**.

[in, out] boolean UseISO14443B

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO14443B**.

[in, out] boolean UseISO15693

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO15693**.

[in, out] boolean UseICode1

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **Philips ICode1**.

[in, out] XID_eCodedTimeSlots in_CodedTimeSlots

Количество временных интервалов, отводимое под выполнение операции с картами.

[in, out] boolean in_AutoSwitchRF

Если **true**, то поле считывателя будет автоматически включаться и выключаться при опросе меток.

По умолчанию **true**.

5.1.4.5.1.2 Методы

do_FullAnticollision()

Запуск выполнения команды/функции **FullAnticollision**.

5.1.4.5.2 Read Data From Cards

IXID_Cmd_Complex_ReadDataFromCards ReadDataFromCards

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.ComplexFuncs.ReadGataFromCards**

Функция **ReadDataFromCards** предназначена для автоматизированного чтения данных карт различных стандартов при помощи указанных считывателей.

Включение и выключение поля считывателя функция выполняет автоматически.

5.1.4.5.2.1 Свойства

[in, out] XID_DevicesList Readers

Список считывателей, которые должны будут участвовать в работе с метками.

Перед выполнением функции, **Readers** должны быть подключены к текущему объекту **XID_ControlX** посредством вызова функции **LinkDevices**.

[in, out] XID_TagsList tags

Список меток, из которого будут браться исходные данные и куда будут помещены результаты выполнения.

[out] IXID_UseStandards in_Standards

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно установить стандарты меток, над которыми следует выполнять операции.

Свойства **in_Standards** описаны в подразделах данного раздела.

[in, out] boolean UseISO14443B

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO14443B**.

[in, out] boolean UseISO14443A

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта

ISO14443A.

[in, out] boolean UseISO15693

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO15693**.

[in, out] boolean UseICode1

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **Philips ICode1**.

[in, out] boolean UseEPC_G1C1HF

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **EPC, Gen1, Class1, 13.56MHz**.

[in, out] XID_eCodedTimeSlots in_CodedTimeSlots

Количество временных интервалов, отводимое под выполнение операции с картами.

[in, out] boolean in_AutoSwitchRF

Если **true**, то поле считывателя будет автоматически включаться и выключаться при опросе меток.

По умолчанию **true**.

5.1.4.5.2.2 Методы

do_ReadDataFromCards()

Запуск выполнения команды/функции **ReadDataFromCards**

5.1.4.5.3 Write Data To Cards

IXID_Cmd_Complex_WriteDataToCards WriteDataToCards

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.ComplexFuncs. WriteDataToCards**

Функция **WriteDataToCards** предназначена для автоматизированной записи данных в карты различных стандартов при помощи указанных считывателей.

Включение и выключение поля считывателя функция выполняет автоматически.

5.1.4.5.3.1 Свойства

[in, out] XID_DevicesList Readers

Список считывателей, которые должны будут участвовать в работе с метками.

Перед выполнением функции, **Readers** должны быть подключены к текущему объекту **XID_ControlX** посредством вызова функции **LinkDevices**.

[in, out] XID_TagsList tags

Список меток, из которого будут браться исходные данные и куда будут помещены результаты выполнения.

[out] IXID_UseStandards in_Standards

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно установить стандарты меток, над которыми следует выполнять операции.

Свойства **in_Standards** описаны в подразделах данного раздела.

[in, out] boolean UseISO15693

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO15693**.

[in, out] boolean UseISO14443A

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO14443A**.

[in, out] boolean UseISO14443B

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO14443B**.

[in, out] boolean UseEPC_G1C1HF

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **EPC, Gen1, Class1, 13.56MHz**.

[in, out] boolean UseICode1

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **Philips ICode1**.

[in, out] XID_eCodedTimeSlots in_CodedTimeSlots

Количество временных интервалов, отводимое под выполнение операции с картами.

[in, out] string in_DataAsTextString

Данные для записи в виде символьной строки.

Если **in_DataAsTextString** = "", то данные для записи будут браться из списка меток для каждой метки соответственно.

[in, out] string in_DataAsHexString

Данные для записи в виде строки шестнадцатеричных символов.

Если **in_DataAsHexString** = "", то данные для записи будут браться из списка меток для каждой метки соответственно.

[in, out] short in_StartBlockShift

Блок, с которого требуется начать запись

[in, out] short in_BlocksCount

Количество записываемых блоков

[in, out] short in_BlockSize

© 2011 MicroEM

Размер блока в байтах

[in, out] boolean in_AutoSwitchRF

Если **true**, то поле считывателя будет автоматически включаться и выключаться при опросе меток.

По умолчанию **true**.

5.1.4.5.3.2 Методы

do_WriteDataToCards()

Запуск выполнения данной команды/функции.

5.1.4.5.4 Типы

5.1.4.5.4.1 XID_eCodedTimeSlots

enum XID_eCodedTimeSlots

Данный перечислимый тип объединяет возможные значения количества временных интервалов, выделяемых для ответа метки:

xidTimeSlots_1	1 временной интервал
xidTimeSlots_4	4 временных интервала
xidTimeSlots_8	8 временных интервалов
xidTimeSlots_16	16 временных интервалов
xidTimeSlots_32	32 временных интервала
xidTimeSlots_64	64 временных интервала
xidTimeSlots_128	128 временных интервалов
xidTimeSlots_256	256 временных интервалов
xidTimeSlots_512	512 временных интервалов

5.1.4.6 Функции командных запросов (Descrete Query Functions)

IXID_DeviceTypes QueryDevices

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices**.

Интерфейс **QueryDevices** объединяет группы команд, относящихся к работе с устройствами

различных типов.

5.1.4.6.1 Общие функции устройств (Any Device Functions)

IXID_DeviceFuncs AnyType

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.AnyType**.

Здесь собраны команды, предназначенные для работы с устройствами любых типов.

5.1.4.6.1.1 ReadConfig

IXID_Cmd_Device_ReadConfig ReadConfig

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.AnyType.ReadConfig**.

Чтение байт конфигурации устройства

[in, out] short in_ConfigLength

Требуемое количество байт конфигурации

[in, out] string out_ConfigData

Данные конфигурации в виде строки шестнадцатеричных символов

do_ReadConfig()

Запуск выполнения данной команды/функции.

5.1.4.6.1.2 WriteConfig

IXID_Cmd_Device_WriteConfig WriteConfig

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.AnyType.WriteConfig**.

Запись байт конфигурации устройства

[in, out] short in_ConfigLength

Требуемое количество байт конфигурации

[in, out] string in_ConfigData

Данные конфигурации в виде строки шестнадцатеричных символов

do_WriteConfig()

Запуск выполнения данной команды/функции.

5.1.4.6.1.3 ReadBusAddress

IXID_Cmd_Device_ReadBusAddress ReadBusAddress

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.AnyType. ReadBusAddress**.

Чтение байта адреса устройства на шине RS485.

[in, out] byte out_BusAddress

Байт адреса устройства на шине RS485.

do_ReadBusAddress()

Запуск выполнения данной команды/функции.

5.1.4.6.1.4 WriteBusAddress

IXID_Cmd_Device_WriteBusAddress WriteBusAddress

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.AnyType. WriteBusAddress**.

Запись байта адреса устройства на шине RS485.

[in, out] byte in_BusAddress

Байт адреса устройства на шине RS485.

do_WriteBusAddress()

Запуск выполнения данной команды/функции.

5.1.4.6.1.5 Signal

IXID_Cmd_Device_Signal Signal

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.AnyType.Signal**.

Подача светового и/или звукового сигнала устройством

[in, out] XID_eSignalTypes in_SignalType

Текущий язык для выдачи сообщений библиотеки, а также описания ошибки.

[in, out] byte in_SignalLength

Время сигнала в десятках миллисекунд

[in, out] byte in_SignalsCount

Количество сигналов

[in, out] XID_eLedColor in_LedColor

Цвет светового сигнала

[in, out] boolean in_LedBlink

Включение/выключение мигания светодиода

© 2011 MicroEM

[in, out] XID_eLedColor in_RemainingColor

Цвет светодиода после светового сигнала

do_Signal()

Запуск выполнения данной команды/функции.

5.1.4.6.1.6 UpdateFirmware

IXID_Cmd_Device_UpdateFirmware UpdateFirmware

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.AnyType.UpdateFirmware**

Обновление программной зашивки устройства

[in, out] string in_FirmwareFileName

Имя файла с прошивкой (*.enc)

[in, out] boolean in_RawUpdate

Включение/выключение режима "сырого" обновления

do_UpdateFirmware()

Запуск выполнения данной команды/функции.

5.1.4.6.1.7 Типы

enum XID_eSignalTypes

Данный перечислимый тип объединяет типы сигналов:

xidSignalLight	Световой сигнал
xidSignalSound	Звуковой сигнал

xidSignalLightAndSound	Световой и звуковой сигнал одновременно

enum XID_eLedColor

Данный перечислимый тип объединяет цвета для световых сигналов:

dnrLedColorRed	Красный
dnrLedColorGreen	Зеленый
dnrLedColorYellow	Желтый

5.1.4.6.2 Функции считывателя (Reader Functions)

IXID_ReaderFuncs Reader

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader**.

Здесь собраны команды, предназначенные для работы с устройствами типа "Считыватель ".

5.1.4.6.2.1 ResetRF

IXID_Cmd_Reader_ResetRF ResetRF

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.ResetRF**.

Сброс поля считывателя на указанное число миллисекунд

[in, out] byte in_ResetTime

Время сброса поля в десятках миллисекунд (если == 0, поле может не включиться снова)

do_ResetRF()

Запуск выполнения данной команды/функции.

5.1.4.6.2.2 RF_ON

IXID_Cmd_Reader_RF_ON RF_ON

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.RF_ON**.

Включение поля считывателя

do_RF_ON()

Запуск выполнения данной команды/функции.

5.1.4.6.2.3 RF_OFF

IXID_Cmd_Reader_RF_OFF RF_OFF

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.RF_OFF**.

Выключение поля считывателя

do_RF_OFF()

Запуск выполнения данной команды/функции.

5.1.4.6.2.4 GetKeysState

IXID_Cmd_Reader_GetKeysState GetKeysState

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.GetKeysState**.

Получение текущего состояния клавиш на считывателе

[in, out] byte out_KeysState

Состояние клавиш считывателя (8 битовых флагов для соответствующего числа клавиш на считывателе)

do_GetKeysState()

Запуск выполнения данной команды/функции.

5.1.4.6.2.5 SetSignalState

IXID_Cmd_Reader_SetSignalState SetSignalState

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.SetSignalState**

Установка состояния органов индикации на считывателе

[in, out] byte in_NewSignalState

Устанавливаемое состояние элементов индикации

do_SetSignalState()

Запуск выполнения данной команды/функции.

5.1.4.6.2.6 GetNoiseFactor

IXID_Cmd_Reader_GetNoiseFactor GetNoiseFactor

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.GetNoiseFactor**.

Получение текущего уровня шума на антенне считывателя

[in, out] short out_NoiseFactor

Фактор шума

do_GetNoiseFactor()

Запуск выполнения данной команды/функции.

© 2011 MicroEM

5.1.4.6.2.7 Функции по стандартам меток (Functions By Tags Standards)

IXID_TagStandards TagStandards

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards**.

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками различных стандартов.

IXID_Std_EPC_Global EPC_Global

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.EPC_Global**.

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками линейки стандартов от **EPC Global**.

IXID_Std_EPC_Gen1 Gen1

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.EPC_Global.Gen1**.

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками подгруппы стандартов **Generation I** от **EPC Global**.

IXID_Std_EPC_Gen1_Class1 Class1

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.EPC_Global.Gen1.Class1**.

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками подгруппы стандартов **Class 1** подгруппы **Generation I** от **EPC Global**.

IXID_Std_EPCFuncs HF

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.EPC_Global.Gen1.Class1.HF**.

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками подгруппы **Class 1** подгруппы **Generation I** от **EPC Global**, работающими на частоте 13.56МГц.

© 2011 MicroEM

IXID_Cmd_Reader_EPC_Anticollision Anticollision

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.EPC_Global.Gen1.Class1.HF.Anticollision**.

Процедура антиколлизии - поиск меток в поле

[in, out] byte in_HashValue

Хеш-значение, используемое при антиколлизии

[in, out] XID_eCodedTimeSlots in_CodedTimeSlots

Количество временных интервалов, отводимое под выполнение операции с картами

[in, out] string in_Mask

Маска, при помощи которой происходит выделение подгруппы серийных номеров меток - задается в виде строки шестнадцатеричных символов

[in, out] byte in_MaskLengthInBits

Количество бит из заданной маски, которое требуется использовать при антиколлизии

[in, out] short out_TimeSlots

Количество временных интервалов, которое было выделено для выполнения текущей операции

[in, out] XID_eSlotErrorCodes out_TimeSlotsStatuses[short TimeSlotIndex]

Коды ошибок, относящиеся к временному интервалу **TimeSlotIndex**

[in, out] string out_EPCs[short TimeSlotIndex]

Полученный **EPC**, относящийся к временному интервалу **TimeSlotIndex -** задается в виде

строки шестнадцатеричных символов

do_Anticollision()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_EPC_WriteBlock WriteBlock

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.EPC_Global.Gen1.Class1.HF.WriteBlock**

Запись блока данных в метки

[in, out] byte in_BlockAddress

Блок, с которого требуется начать запись

[in, out] byte in_Block

Данные блока (1 байт для **EPC**)

do_WriteBlock()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_EPC_Destroy Destroy

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.EPC_Global.Gen1.Class1.HF.Destroy**.

Деактивация метки

[in, out] string in_EPC

EPC метки, которую требуется деактивировать - задается в виде строки шестнадцатеричных

символов

[in, out] string in_DestroyCode

Код деактивации метки - задается в виде строки шестнадцатеричных символов

do_Descroy()

Запуск выполнения данной команды/функции.

IXID_Std_ISO ISO

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO**.

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками линейки стандартов от **International Standard Organization (ISO)**.

IXID_Std_ISO15693Funcs ISO15693

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO15693**.

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками стандарта **ISO15693** от **ISO**.

IXID_Cmd_Reader_ISO15693_Anticollision Anticollision

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO15693.Anticollision**

Процедура антиколлизии - поиск меток в поле

[in, out] boolean in_Use1TimeSlot

Флаг, задающий, стоит ли использовать один временной интервал (в противном случае будет

использовано 16 интервалов)

[in, out] boolean in_AddAFI

Флаг, задающий, стоит ли при выполнении антиколлизии использовать значение **AFI (Application Family Identifier)**

[in, out] byte in_AFI

Идентификатор семейства приложения **AFI (Application Family Identifier)**

[in, out] byte in_MaskLengthInBits

Количество бит из заданной маски, которое требуется использовать при антиколлизии

[in, out] string in_Mask

Маска, при помощи которой происходит выделение подгруппы серийных номеров меток - задается в виде строки шестнадцатеричных символов.

[in, out] short out_TimeSlots

Количество временных интервалов, которое было выделено для выполнения текущей операции

[in, out] array of XID_eSlotErrorCodes out_arrTimeSlotsStatuses

Массив кодов ошибок

[in, out] XID_eSlotErrorCodes out_TimeSlotsStatuses[short TimeSlotIndex]

Коды ошибок, относящиеся к временному интервалу **TimeSlotIndex**

[in, out] string array out_arrUIDs

Массив полученных **UID** - в виде строк шестнадцатеричных символов

[in, out] string out_UIDs[byte TimeSlotIndex]

Полученный **UID**, относящийся к временному интервалу **TimeSlotIndex -** в виде строки шестнадцатеричных символов

do_Anticollision()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO15693_GetBlocksSecurities GetBlocksSecurities

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.ISO.ISO15693.GetBlocksSecurities**

Получение статусов защищенности блоков от записи

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[in, out] byte in_BlockAddress

Начальный блок для выполнения данной операции

[in, out] byte in_BlocksCount

Количество блоков, над которыми требуется выполнить данную операцию (начиная с **in_BlockAddress**)

[out] IXID_ISO15693_Security out_BlocksSecurities[byte BlockIndex]

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно получить с татус защиты для блока **BlockIndex**

do_GetBlocksSecurities()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO15693_GetSystemInfo GetSystemInfo

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.ISO.ISO15693.GetSystemInfo**

Получение системной информации метки

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[out] IXID_ISO15693_SystemInfo out_SystemInformation

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно получить считанную с метки системную информацию

[in, out] boolean SupportAFI

Флаг, сигнализирующий о том, поддерживает ли данная метка **AFI** или нет

[in, out] boolean SupportDSFID

Флаг, сигнализирующий о том, поддерживает ли данная метка **DSFID** или нет

[in, out] boolean HasMemSizeInfo

Флаг, сигнализирующий о том, поддерживает ли данная метка информацию о структуре памяти или нет

[in, out] boolean HasICreferenceInfo

Флаг, сигнализирующий о том, поддерживает ли данная метка **IC reference**

[in, out] string RawInfoAsHexText

Системная информация в виде строки шестнадцатеричных символов

[in, out] byte AFI

Идентификатор семейства приложения **AFI (Application Family Identifier)**

[in, out] byte DSFID

Идентификатор формата хранилища данных **DSFID (Data Storage Format Identifier)**

[out] IXID_ISO15693_SystemMemoryInfo MemoryInfo

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно получить считанную с метки информацию об устройстве памяти

[in, out] short BlockSize

Размер блока в байтах

[in, out] short NumberOfBlocks

Количество блоков в метке

© 2011 MicroEM

[in, out] byte IC_ReferenceInfo

Определяемая поизводителем дополнительная информация о микросхеме

do_GetSystemInfo()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO15693_LockAFI LockAFI

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO15693.LockAFI**.

Блокировка перезаписи байта **AFI**

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[in, out] boolean in_WaitForEOF

Флаг, задающий, стоит ли при выполнении записи ждать **EOF** от метки (для чипов от **Texas Instruments** должно быть **True**, для **Philips - False**)

do_LockAFI()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO15693_LockBlock LockBlock

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO15693.LockBlock**.

Блокировка перезаписи блока

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[in, out] boolean in_WaitForEOF

Флаг, задающий, стоит ли при выполнении записи ждать **EOF** от метки (для чипов от **Texas Instruments** должно быть **True**, для **Philips - False**)

[in, out] byte in_BlockAddress

Начальный блок для выполнения данной операции

do_LockBlock()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO15693_LockDSFID LockDSFID

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO15693.LockDSFID**.

Блокировка перезаписи байта **DSFID**

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[in, out] boolean in_WaitForEOF

Флаг, задающий, стоит ли при выполнении записи ждать **EOF** от метки (для чипов от **Texas Instruments** должно быть **True**, для **Philips - False**)

do_LockDSFID()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO15693_ReadBlock ReadBlock

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.ISO.ISO15693.ReadBlock**

Чтение блока данных метки

[in, out] boolean in_AddSecurityByte

Флаг, задающий, стоит ли помимо данных блока читать также информацию о защите блока

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки

шестнадцатеричных символов

[in, out] byte in_BlockAddress

Начальный блок для выполнения данной операции

[in, out] byte out_BlockSize

Размер блока в байтах

[in, out] string out_BlockAsHexString

Данные блока в виде строки шестнадцатеричных символов

[in, out] string out_BlockAsTextString

Данные блока в виде символьной строки

[out] IXID_ISO15693_Security out_BlockSecurity

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно получить статус защиты для блока

do_ReadBlock()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO15693_ReadBlocks ReadBlocks

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO15693.ReadBlocks**

Чтение блоков данных

[in, out] boolean in_AddSecurityByte

Флаг, задающий, стоит ли помимо данных блока читать также информацию о защите блока

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[in, out] byte in_BlockAddress

Начальный блок для выполнения данной операции

[in, out] byte in_BlocksCount

Количество блоков, над которыми требуется выполнить данную операцию (начиная с **in_BlockAddress**)

[in, out] byte out_BlockSize

Размер блока в байтах

[in, out] string array out_arrBlocksAsHexStrings

Массив полученных блоков - в виде строк шестнадцатеричных символов

[in, out] string out_BlocksAsHexStrings[byte BlockIndex]

Данные блока **BlockIndex** в виде строки шестнадцатеричных символов

© 2011 MicroEM

[in, out] string array out_arrBlocksAsTextStrings

Массив полученных блоков - в виде символьных строк

[in, out] string out_BlockAsTextString[byte BlockIndex]

Данные блока **BlockIndex** в виде символьной строки

[out] IXID_ISO15693_Security out_BlocksSecurities[byte BlockIndex]

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно получить с татус защиты для блока **BlockIndex**

do_ReadBlocks()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO15693_ResetToReady ResetToReady

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO15693.ResetToReady**.

Возврат метки в состояние **Ready**

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

© 2011 MicroEM

do_ResetToReady()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO15693_Select Select

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO15693.Select**

Выбор метки для дальнейшей работы

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

do_Select()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO15693_StayQuiet StayQuiet

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO15693.StayQuiet**

Перевод метки в состояние **Quiet**

В этом состоянии метка не будет отвечать на команды, кроме **WakeUp**

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

© 2011 MicroEM

do_StayQuiet()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO15693_WriteAFI WriteAFI

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO15693.WriteAFI**

Запись байта **AFI**

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[in, out] boolean in_WaitForEOF

Флаг, задающий, стоит ли при выполнении записи ждать **EOF** от метки (для чипов от **Texas Instruments** должно быть **True**, для **Philips - False**)

[in, out] byte in_AFI

Идентификатор семейства приложения **AFI (Application Family Identifier)**

do_WriteAFI()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO15693_WriteBlock WriteBlock

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO15693.WriteBlock**

Запись блока данных

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[in, out] boolean in_WaitForEOF

Флаг, задающий, стоит ли при выполнении записи ждать **EOF** от метки (для чипов от **Texas Instruments** должно быть **True**, для **Philips - False**)

[in, out] byte in_BlockAddress

Начальный блок для выполнения данной операции

[in, out] byte in_BlockSize

Размер блока в байтах

[in, out] string in_BlockAsHexString

Данные блока в виде строки шестнадцатеричных символов

[in, out] string in_BlockAsTextString

Данные блока в виде символьной строки

© 2011 MicroEM

do_WriteBlock()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO15693_WriteBlocks WriteBlocks

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO15693.WriteBlocks**

Запись блоков данных

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[in, out] boolean in_WaitForEOF

Флаг, задающий, стоит ли при выполнении записи ждать **EOF** от метки (для чипов от **Texas Instruments** должно быть **True**, для **Philips - False**)

[in, out] byte in_BlockAddress

Начальный блок для выполнения данной операции

[in, out] byte in_BlocksCount

Количество блоков, над которыми требуется выполнить данную операцию (начиная с **in_BlockAddress**)

© 2011 MicroEM

[in, out] byte in_BlockSize

Размер блока в байтах

[in, out] string in_BlocksAsHexStrings[byte BlockIndex]

Данные блока **BlockIndex** в виде строки шестнадцатеричных символов

[in, out] string in_BlockAsTextString[byte BlockIndex]

Данные блока **BlockIndex** в виде символьной строки

do_WriteBlocks()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO15693_WriteDSFID WriteDSFID

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO15693.WriteDSFID**

Запись байта **DSFID**

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[in, out] boolean in_WaitForEOF

Флаг, задающий, стоит ли при выполнении записи ждать **EOF** от метки (для чипов от **Texas Instruments** должно быть **True**, для **Philips - False**)

[in, out] byte in_DSFID

Идентификатор формата хранилища данных **DSFID (Data Storage Format Identifier)**

do_WriteDSFID()

Запуск выполнения данной команды/функции.

enum XID_eIS015693_QueryTypes

Данный перечислимый тип объединяет возможные значения видов обращения к меткам:

xidISO15693_QueryAll	обращение ко всем активным меткам в поле
xidISO15693_QueryByUID	обращение к метке с заданным **UID**
xidISO15693_QuerySelected	обращение к последней выбранной командой **Select** метке

IXID_ISO15693_Security BlockSecurity

Представляет собой указатель на интерфейс типа **IUnknown**, посредством которого можно получить считанную с метки информацию о статусе защиты блока

[in, out] byte RawByte

Байт статуса защищенности в "сыром" виде

[in, out] boolean Locked

Признак блокировки блока (**True** - заблокирован, **False** - нет)

© 2011 MicroEM

IXID_Std_ISO14443AFuncs ISO14443A

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO14443A**

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками стандарта **ISO14443A** от **ISO**.

IXID_Cmd_Reader_ISO14443A_ActivateIdle ActivateIdle

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO14443A.ActivateIdle**

Выделение очередной метки из поля считывателя

[in, out] string out_UID

Полученный **UID -** в виде строки шестнадцатеричных символов

[in, out] boolean out_Is_ISO14443_4_Compatible

Признак совместимости метки со стандартом **ISO14443-4**

do_ActivateIdle()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO14443A_Halt Halt

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO14443A.Halt**

Перевод последней выделенной метки в состояние **Halt**

В этом состоянии метка не будет отвечать на команды, кроме **WakeUp**

do_Halt()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO14443A_WakeUp WakeUp

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO14443A.WakeUp**

Вывод метки из состояния **Halt**

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[in, out] boolean out_Is_ISO14443_4_Compatible

Признак совместимости метки со стандартом **ISO14443-4**

do_WakeUp()

Запуск выполнения данной команды/функции.

IXID_Std_ISO14443BFuncs ISO14443B

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO14443B**.

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками стандарта **ISO14443B** от **ISO**.

IXID_Cmd_Reader_ISO14443B_ActivateIdle ActivateIdle

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO14443B.ActivateIdle**

Выделение очередной метки из поля считывателя

[in, out] byte in_AppFamilyID

Идентификатор семейства приложения **(Application Family Identifier)**

[in, out] XID_eISO14443B_SlotsCount in_SlotsCount

Количество временных интервалов, отводимое под выполнение операции с картами

[in, out] byte out_CardsCount

Количество найденных в поле карт

[out] IXID_ISO14443B_Card out_GetCardInfo[byte CardIndex]

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно получить информацию о прочитанной карте с индексом **CardIndex**

do_ActivateIdle()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO14443B_Attrib Attrib

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO14443B.Attrib**.

Установка/получение атрибутов метки

[in, out] string in_PUPI

PUPI метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[out] IXID_ISO14443B_Param1 in_Param1

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно установить параметр № 1 данной команды

[in, out] XID_eISO14443B_MinimumTR0 MinimumTR0

Minimum TR0 indicates to the PICC the minimum delay before responding after the end of a command sent by a
PCD. The default value has been defined in ISO/IEC 14443-2 clause 9.2.5.

NOTE Minimum TR0 is required by the PCD when switching from transmit to receive and its value depends on the PCD
performance.

[in, out] XID_eISO14443B_MinimumTR1 MinimumTR1

Minimum TR1 indicates to the PICC the minimum delay between subcarrier modulation start and beginning of data
transmission. The default value has been defined in ISO/IEC 14443-2 clause 9.2.5.

NOTE Minimum TR1 is required by the PCD for synchronization with the PICC and its value depends on the PCD
performance.

[in, out] boolean RequiredSOF

indicate the PCD capability to support suppression of the SOF from PICC to PCD, which
may reduce communication overhead. The suppression of SOF is optional for the PICC.

[in, out] boolean RequiredEOF

indicate the PCD capability to support suppression of the EOF from PICC to PCD, which
may reduce communication overhead. The suppression of EOF is optional for the PICC.

[out] IXID_ISO14443B_Param2 in_Param2

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно установить параметр № 2 данной команды

© 2011 MicroEM

[in, out] XID_eISO14443B_MaxFrameSize MaxFrameSize

maximum frame size that can be received by the PCD

[in, out] XID_eISO14443B_BitRate PCD_to_PICC_BitRate

[in, out] XID_eISO14443B_BitRate PICC_to_PCD_BitRate

[out] IXID_ISO14443B_Param3 in_Param3

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно установить параметр № 3 данной команды

[in, out] XID_eISO14443B_ProtocolType ProtocolType

Тип поддерживаемого микросхемой метки протокола

[out] IXID_ISO14443B_Param4 in_Param4

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно установить параметр № 4 данной команды

[in, out] byte CardIdentifier

Card Identifier (CID) defines the logical number of the addressed PICC in the range from 0 to 14. The value 15 is RFU. The CID is specified by the PCD and shall be unique for each active PICC. If the PICC does not support CID, code value (0000)b shall be used

[in, out] array of byte inout_arrHigerLevelINF

Массив байт

Any higher layer command transferable as the INF field of ISO/IEC 14443-4 may be included.

It is not mandatory for the PICC to process successfully any command in this context.
The PICC shall however process successfully such message if no application command is included.

[in, out] byte out_MaxBufLenIndex

Maximum Buffer Length Index (MBLI). It is used by the PICC to let the PCD know the limit of its internal buffer to received chained frames.
The coding of MBLI is as follows :
s MBLI = 0 means that the PICC provides no information on its internal input buffer size ;
s MBLI > 0 is used to calculate the actual internal maximum buffer length (MBL) according to the following
formula : MBL = (PICC Maximum Frame Size)* 2(MBLI-1) where the PICC maximum frame size is returned
by the PICC in its ATQB. When it send chained frames to a PICC, the PCD shall ensure that the accumulated length is never greater than MBL.

[in, out] byte out_CardID

CID карты

Если карта не поддерживает **CID**, возвращается 0

do_Attrib()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO14443B_Halt Halt

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO14443B.Halt**

Перевод последней выделенной метки в состояние **Halt**

В этом состоянии метка не будет отвечать на команды, кроме **WakeUp**

[in, out] string in_PUPI

PUPI метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

do_Halt()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ISO14443B_WakeUp WakeUp

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.ISO.ISO14443B.WakeUp**

Вывод метки из состояния **Halt**

[in, out] byte in_AppFamilyID

Идентификатор семейства приложения **(Application Family Identifier)**

[in, out] XID_eISO14443B_SlotsCount in_SlotsCount

Количество временных интервалов, отводимое под выполнение операции с картами

[in, out] byte out_CardsCount

Количество найденных в поле карт

[out] IXID_ISO14443B_Card out_GetCardInfo[byte CardIndex]

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно получить информацию о прочитанной карте с индексом **CardIndex**

do_WakeUp()

Запуск выполнения данной команды/функции.

enum XID_eISO14443B_SlotsCount

Данный перечислимый тип объединяет возможные значения количества временных интервалов:

xidISO14443B_SlotsCount_1	1 временной интервал
xidISO14443B_SlotsCount_2	2 временных интервала
xidISO14443B_SlotsCount_4	4 временных интервала
xidISO14443B_SlotsCount_8	8 временных интервалов
xidISO14443B_SlotsCount_16	16 временных интервалов

enum XID_eISO14443B_MinimumTR0

Данный перечислимый тип объединяет возможные минимальные значения **TR0**:

xidISO14443B_MinimumTR0_Default	значение по умолчанию
xidISO14443B_MinimumTR0_48divFS	48 / FS
xidISO14443B_MinimumTR0_16divFS	16 / FS
xidISO14443B_MinimumTR0_RFU	зарезервировано

enum XID_eISO14443B_MinimumTR1

Данный перечислимый тип объединяет возможные минимальные значения **TR1**:

xidISO14443B_MinimumTR1_Default	значение по умолчанию
xidISO14443B_MinimumTR1_64divFS	64 / FS
xidISO14443B_MinimumTR1_16divFS	16 / FS
xidISO14443B_MinimumTR1_RFU	зарезервировано

enum XID_eISO14443B_MaxFrameSize

Данный перечислимый тип объединяет возможные максимальные размеры кадра:

xidISO14443B_MaxFrameSize_16_Bytes	16 байт
xidISO14443B_MaxFrameSize_24_B	24 байта

© 2011 MicroEM

ytes	
xidISO14443B_MaxFrameSize_32_B	32 байта
ytes	
xidISO14443B_MaxFrameSize_40_B	40 байт
ytes	
xidISO14443B_MaxFrameSize_48_B	48 байт
ytes	
xidISO14443B_MaxFrameSize_64_B	64 байта
ytes	
xidISO14443B_MaxFrameSize_96_B	96 байт
ytes	
xidISO14443B_MaxFrameSize_128_	128 байт
Bytes	
xidISO14443B_MaxFrameSize_256_	256 байт
Bytes	
xidISO14443B_MaxFrameSize_RFU1	зарезервировано
xidISO14443B_MaxFrameSize_RFU2	зарезервировано
xidISO14443B_MaxFrameSize_RFU3	зарезервировано
xidISO14443B_MaxFrameSize_RFU4	зарезервировано
xidISO14443B_MaxFrameSize_RFU5	зарезервировано
xidISO14443B_MaxFrameSize_RFU6	зарезервировано
xidISO14443B_MaxFrameSize_RFU7	зарезервировано

enum XID_eISO14443B_BitRate

Данный перечислимый тип объединяет возможные значения скорости обмена:

xidISO14443B_BitRate_106_kbps	106 килобит в секунду
xidISO14443B_BitRate_212_kbps	212 килобит в секунду
xidISO14443B_BitRate_424_kbps	424 килобита в секунду
xidISO14443B_BitRate_847_kbps	847 килобит в секунду

enum XID_eISO14443B_ProtocolType

Данный перечислимый тип объединяет возможные значения типа протокола:

xidISO14443B_ProtocolType_Not_CompliantWithISO14443_4	протокол не совместимо со стандартом **ISO14443-4**
xidISO14443B_ProtocolType_CompliantWithISO14443_4	протокол совместимо со стандартом **ISO14443-4**

IXID_ISO14443B_Card Card

Представляет собой указатель на интерфейс типа **IDispatch**, посредством которого можно получить информацию о карте

[in, out] string ProtocolInfo

The Protocol Info field indicates the parameters supported by the PICC

- задается в виде строки шестнадцатеричных символов

[in, out] string ApplicationData

The Application data field is used to inform the PCD which applications are currently installed in the PICC. This
information allows the PCD to select the desired PICC in the presence of more than one PICC.
The application data is defined dependent upon the ADC (Application Data Coding) field in the Protocol Info (see
7.9.4.), which defines if either the CRC_B compressing method described below or proprietary coding is used.
When the CRC_B compressing coding is used, Application Data field contains the following :
1st byte 2nd, 3rd bytes 4th byte
AFI
(1 byte)
CRC_B(AID)
(2 bytes)
Numbers of Applications
(1 byte)
MSB LSB MSB LSB MSB LSB
Figure 24 — Application Data format

[in, out] string PUPI

PUPI метки - задается в виде строки шестнадцатеричных символов

IXID_Std_Philips NXP

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.NXP**.

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками линейки стандартов от **NXP**.

IXID_Std_Philips_ICode ICode

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.ICode**.

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками серии стандартов **ICode** от **NXP**.

IXID_Std_ICodeFuncs ICode1

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.ICode.ICode1**

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками стандарта **ICode I** от **Philips**.

IXID_Cmd_Reader_ICode_Anticollision Anticollision

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.ICode.ICode1.Anticollision**

Выполнение процедуры антиколлизии - поиск карт в поле

[in, out] byte in_HashValue

Хеш-значение, используемое при антиколлизии

[in, out] byte in_FamilyCode

Идентификатор кода семейства

[in, out] byte in_ApplicationIdentifier

Идентификатор приложения

[in, out] XID_eCodedTimeSlots in_CodedTimeSlots

Количество временных интервалов, отводимое под выполнение операции с картами

[in, out] short out_TimeSlots

Количество временных интервалов, которое было выделено для выполнения текущей операции

[in, out] XID_eSlotErrorCodes out_TimeSlotsStatuses[short TimeSlotIndex]

Коды ошибок, относящиеся к временному интервалу **TimeSlotIndex**

[in, out] string out_IDs[short TimeSlotIndex]

Полученный **ID**, относящийся к временному интервалу **TimeSlotIndex -** задается в виде строки шестнадцатеричных символов

do_Anticollision()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ICode_EAS EAS

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.Philips.ICode.ICode1.EAS**

Определение наличия меток в поле (специализированная функция для антикражевой системы)

[in, out] byte in_FamilyCode

Идентификатор кода семейства

[in, out] byte in_ApplicationIdentifier

Идентификатор приложения

do_EAS()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ICode_ReadSelected ReadSelected

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.ICode.ICode1.ReadSelected**.

Чтение блоков данных меток, находящихся в состоянии **Selected**, то есть разложившихся без коллизий в слоты при выполнении команды **Anticollision**

[in, out] byte in_BlockAddress

Начальный блок для выполнения данной операции

[in, out] byte in_BlocksCount

Количество блоков, над которыми требуется выполнить данную операцию (начиная с **in_BlockAddress**)

[in, out] short out_TimeSlots

Количество временных интервалов, которое было выделено для выполнения текущей операции

[in, out] XID_eSlotErrorCodes out_TimeSlotsStatuses[short TimeSlotIndex]

Коды ошибок, относящиеся к временному интервалу **TimeSlotIndex**

[in, out] string out_BlockAsTextString[byte BlockIndex]

Данные блока **BlockIndex** в виде символьной строки

[in, out] string out_BlocksAsHexStrings[byte BlockIndex]

Данные блока **BlockIndex** в виде строки шестнадцатеричных символов

do_ReadSelected()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ICode_ReadUnselected ReadUnselected

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.Philips.ICode.ICode1.ReadUnselected**

Чтение блоков данных меток, находящихся в состоянии **Unselected**, то есть не разложившихся без коллизий в слоты при выполнении команды **Anticollision**

[in, out] byte in_HashValue

Хеш-значение, используемое при антиколлизии

[in, out] byte in_FamilyCode

Идентификатор кода семейства

[in, out] byte in_ApplicationIdentifier

Идентификатор приложения

[in, out] XID_eCodedTimeSlots in_CodedTimeSlots

Количество временных интервалов, отводимое под выполнение операции с картами

[in, out] byte in_BlockAddress

Начальный блок для выполнения данной операции

[in, out] byte in_BlocksCount

Количество блоков, над которыми требуется выполнить данную операцию (начиная с **in_BlockAddress**)

[in, out] short out_TimeSlots

Количество временных интервалов, которое было выделено для выполнения текущей операции

[in, out] XID_eSlotErrorCodes out_TimeSlotsStatuses[short TimeSlotIndex]

Коды ошибок, относящиеся к временному интервалу **TimeSlotIndex**

[in, out] string out_BlockAsTextString[byte BlockIndex]

Данные блока **BlockIndex** в виде символьной строки

[in, out] string out_BlocksAsHexStrings[byte BlockIndex]

Данные блока **BlockIndex** в виде строки шестнадцатеричных символов

do_ReadUnselected()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ICode_ResetQuietBit ResetQuietBit

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.Philips.ICode.ICode1.ResetQuietBit**

Сброс бита **Quiet** в метках, находящихся в состоянии **Selected**, то есть разложившихся без коллизий в слоты при выполнении команды **Anticollision**

do_ResetQuietBit()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ICode_WriteBlock WriteBlock

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.**

TagStandards.Philips.ICode.ICode1.WriteBlock

Запись блока данных в метки, находящиеся в слотах, указанных в **in_SelectTimeSlot**

[in, out] byte in_HashValue

Хеш-значение, используемое при антиколлизии

[in, out] byte in_BlockAddress

Начальный блок для выполнения данной операции

[in, out] boolean in_SelectTimeSlot[short TimeSlotIndex]

Флаг, указывающий на то, адресована ли эта операция метке, закрепленной за интервалом **TimeSlotIndex**

[in, out] string in_BlockAsTextString

Данные блока в виде символьной строки

[in, out] string in_BlockAsHexString

Данные блока в виде строки шестнадцатеричных символов

[in, out] short out_TimeSlots

Количество временных интервалов, которое было выделено для выполнения текущей операции

[in, out] XID_eSlotErrorCodes out_TimeSlotsStatuses[short TimeSlotIndex]

Коды ошибок, относящиеся к временному интервалу **TimeSlotIndex**

[in, out] string out_IDs[short TimeSlotIndex]

Полученный **ID**, относящийся к временному интервалу **TimeSlotIndex -** задается в виде строки шестнадцатеричных символов

do_WriteBlock()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_ICode_Halt Halt

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.Philips.ICode.ICode1.Halt**

Перевод меток, находящихся в слотах, указанных в **in_SelectTimeSlot**, в состояние **Halt**

[in, out] byte in_HashValue

Хеш-значение, используемое при антиколлизии

[in, out] boolean in_SelectTimeSlot[short TimeSlotIndex]

Флаг, указывающий на то, адресована ли эта операция метке, закрепленной за интервалом **TimeSlotIndex**

[in, out] short out_TimeSlots

Количество временных интервалов, которое было выделено для выполнения текущей операции

[in, out] XID_eSlotErrorCodes out_TimeSlotsStatuses[short TimeSlotIndex]

Коды ошибок, относящиеся к временному интервалу **TimeSlotIndex**

[in, out] string out_IDs[short TimeSlotIndex]

Полученный **ID**, относящийся к временному интервалу **TimeSlotIndex -** задается в виде строки шестнадцатеричных символов

do_Halt()

Запуск выполнения данной команды/функции.

IXID_Std_ICodeSLIFuncs ICodeSLI

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI**

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками стандарта **ICode II (SLI)** от **Philips**.

IXID_Std_ISO15693Funcs ISO15693

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.ICodeSLIISO15693** .

Здесь собраны базовые команды, предназначенные для управления устройствами типа " Считыватель " при работе с метками стандарта **ICode SLI** от **Philips**.

Подробное описание базовых команд для работы с данным стандартом идентично описанию **IXID_Std_ISO15693Funcs**

IXID_ExtPhilips15693_SetEAS SetEAS

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.SetEAS**

Установка бита **EAS** в 1

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

© 2011 MicroEM

[in, out] byte in_ManufacturerCode

Код изготовителя

do_SetEAS()

Запуск выполнения данной команды/функции.

IXID_ExtPhilips15693_ResetEAS ResetEAS

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.ResetEAS**

Установка бита **EAS** в 0

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[in, out] byte in_ManufacturerCode

Код изготовителя

do_ResetEAS()

Запуск выполнения данной команды/функции.

IXID_ExtPhilips15693_LockEAS LockEAS

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.LockEAS**

Блокировка установки бита **EAS**

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[in, out] byte in_ManufacturerCode

Код изготовителя

do_LockEAS()

Запуск выполнения данной команды/функции.

IXID_ExtPhilips15693_EASAlarm EASAlarm

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.EASAlarm**

При условии, что в метке бит **EAS** установлен в 1, метка отвечает последовательностью **EAS**

[in, out] XID_eISO15693_QueryTypes in_SelectionType

Тип выборки карты для выполнения операции

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[in, out] byte in_ManufacturerCode

Код изготовителя

[in, out] string out_EASSequenceAsHexString

Полученная последовательность **EAS** в виде строки шестнадцатеричных символов

do_EASAlarm()

Запуск выполнения данной команды/функции.

IXID_Std_EPCFuncs ICodeEPC_G1C1HF

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeEPC_G1C1HF**.

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками стандарта **ICode EPC** от **Philips**.

Подробное описание команд для работы с данным стандартом идентично описанию **IXID_Std_EPCFuncs**

IXID_Std_Philips_Mifare Mifare

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare**

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками серии стандартов **Mifare** от **NXP**.

IXID_Std_MifareFuncs Standard1k

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.Philips.Mifare.Standard1k**

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками стандарта **Mifare Standard 1k** от **NXP**.

IXID_Std_ISO14443AFuncs ISO14443A

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.Philips.Mifare.Standard1k.ISO14443A**.

Здесь собраны базовые команды, предназначенные для управления устройствами типа " Считыватель " при работе с метками стандарта **Mifare Standard 1k или 4k** от **NXP**.

Подробное описание базовых команд для работы с данным стандартом идентично описанию **IXID_Std_ISO14443AFuncs**

IXID_Cmd_Reader_MifareStandard_Authenticate Authenticate

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.Philips.Mifare.Standard1k.Authenticate**

Аутентификация сектора карты по заданному ключу

[in, out] string in_UID

UID метки, над которой требуется выполнить данную операцию - задается в виде строки шестнадцатеричных символов

[in, out] XID_eMifareStandardKeyType in_KeyType

Тип ключа для выполнения данной процедуры

[in, out] byte in_KeyNumber

Порядковый номер ключа для выполнения данной операции

[in, out] byte in_SectorNumber

Индекс сектора для выполнения данной операции

[in, out] string in_CodedKey

Закодированный ключ в виде строки шестнадцатеричных символов.

do_Authenticate()

Запуск выполнения данной команды/функции.

Свойство **in_CodedKey** не учитывается - ключ прописывается в считыватель отдельно, командой **LoadKey**.

do_AuthenticateWithKey()

Запуск выполнения данной команды/функции.

Свойство **in_CodedKey** учитывается в качестве кодированного ключа (кодируется коммандой **CodeKey**).

IXID_Cmd_Reader_MifareStandard_LoadKey LoadKey

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.Standard1k.LoadKey**

Загрузка ключа в память считывателя

[in, out] XID_eMifareStandardKeyType in_KeyType

Тип ключа для выполнения данной процедуры

[in, out] byte in_KeyNumber

Порядковый номер ключа для выполнения данной операции

[in, out] byte in_SectorNumber

Индекс сектора для выполнения данной операции

[in, out] string in_Key

Кодированный ключ - задаётся в виде строки шестнадцатеричных символов.

Это свойство используется только при вызове метода **do_AuthenticateWithKey**.

do_LoadKey()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareStandard_GenerateKey GenerateKey

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.Standard1k.GenerateKey**

Генерация случайного ключа

[in, out] string out_Key

Сгенерированный ключ в виде строки шестнадцатеричных символов

do_GenerateKey()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareStandard_CodeKey CodeKey

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.Standard1k.CodeKey**

Кодирование ключа

[in, out] string in_Key

© 2011 MicroEM

Кодированный ключ - задаётся в виде строки шестнадцатеричных символов.

Это свойство используется только при вызове метода **do_AuthenticateWithKey**.

[in, out] string out_CodedKey

Полученный закодированный ключ в виде строки шестнадцатеричных символов

do_CodeKey()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareStandard_ReadBlock ReadBlock

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.Standard1k.ReadBlock**

Чтение блока данных

[in, out] byte in_BlockNumber

Номер блока (начиная от 0) для выполнения данной операции

[in, out] string out_BlockAsHexString

Данные блока в виде строки шестнадцатеричных символов

[in, out] string out_BlockAsTextString

Данные блока в виде символьной строки

do_ReadBlock()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareStandard_WriteBlock WriteBlock

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.Standard1k.WriteBlock**

Запись блока данных

[in, out] byte in_BlockNumber

Номер блока (начиная от 0) для выполнения данной операции

[in, out] string in_BlockAsHexString

Данные блока в виде строки шестнадцатеричных символов

[in, out] string in_BlockAsTextString

Данные блока в виде символьной строки

do_WriteBlock()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareStandard_ReadTrailer ReadTrailer

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.Standard1k.ReadTrailer**

Чтение **трейлера** сектора (блок 3)

[in, out] string out_KeyA

© 2011 MicroEM

Считанный ключ A

[in, out] string out_KeyB

Считанный ключ B

[in, out] byte out_Byte9

Считанный байт № 9

[out] IXID_MifareStandard_AccessConditions out_AccessConditions

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно получить/установить параметры доступа к блоку/трейлеру

do_ReadTrailer()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareStandard_WriteTrailer WriteTrailer

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.Standard1k.WriteTrailer**

Запись **трейлера** сектора (блок 3)

[in, out] string in_KeyA

Записываемый ключ A

[in, out] string in_KeyB

Записываемый ключ B

[in, out] byte in_Byte9

Записываемый байт № 9

[out] IXID_MifareStandard_AccessConditions in_AccessConditions

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно получить/установить параметры доступа к блоку/трейлеру

do_WriteTrailer()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareStandard_ReadValue ReadValue

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.Standard1k.ReadValue**

Чтение данных блока как значения

[in, out] byte in_BlockNumber

Номер блока (начиная от 0) для выполнения данной операции

[in, out] long out_Value

Считанное значение

[in, out] byte out_Address

Считанный адрес

do_ReadValue()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareStandard_WriteValue WriteValue

© 2011 MicroEM

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.Standard1k.WriteValue**

Запись данных блока как значения

[in, out] byte in_BlockNumber

Номер блока (начиная от 0) для выполнения данной операции

[in, out] long in_Value

Записываемое значение

[in, out] byte in_Address

Записываемый адрес

do_WriteValue()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareStandard_IncrementValue IncrementValue

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.Standard1k.IncrementValue**

Увеличение значения, записанного в блоке

[in, out] byte in_SourceBlockAddress

Номер исходного блока

[in, out] long in_AddendValue

Прибавляемое значение

[in, out] byte in_DestinationBlockNumber

Номер блока назначения

do_IncrementValue()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareStandard_DecrementValue DecrementValue

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.Standard1k.DecrementValue**

Уменьшение значения, записанного в блоке

[in, out] byte in_SourceBlockAddress

Номер исходного блока

[in, out] long in_SubtrahendValue

Вычитаемое значение

[in, out] byte in_DestinationBlockNumber

Номер блока назначения

do_DecrementValue()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareStandard_CopyValue CopyValue

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.Standard1k.CopyValue**

© 2011 MicroEM

Копирование значения, записанного в блоке

[in, out] byte in_SourceBlockAddress

Номер исходного блока

[in, out] byte in_DestinationBlockNumber

Номер блока назначения

do_CopyValue()

Запуск выполнения данной команды/функции.

IXID_MifareStandard_AccessConditions AccessConditions

Представляет собой указатель на интерфейс типа **IUnknown**, посредством которого можно получить/установить параметры доступа к блоку/трейлеру

IXID_MifareStandard_BlockAccessConditions OfBlock[byte BlockNumber]

Представляет собой указатель на интерфейс типа **IDispatch**, посредством которого можно получить/установить параметры доступа к блоку **BlockNumber**

[in, out] XID_eMifareStandardAccessConditionType AccessConditionType

Задает набор правил доступа

[in, out] XID_eMifareStandardAccessType WriteData

Задает тип доступа для записи

[in, out] XID_eMifareStandardAccessType ReadData

Задает тип доступа для чтения

[in, out] XID_eMifareStandardAccessType IncrementValue

Задает тип доступа для увеличения значения

[in, out] XID_eMifareStandardAccessType DecrementValue

Задает тип доступа для уменьшения значения

[in, out] XID_eMifareStandardAccessType CopyValue

Задает тип доступа для копирования значения

enum XID_eMifareStandardAccessType

Данный перечислимый тип объединяет следующие возможные типы доступа:

xidMifareStandardAccessType_NoAccess	нет доступа
xidMifareStandardAccessType_ByKeyA	доступ по ключу A
xidMifareStandardAccessType_ByKeyB	доступ по ключу B
xidMifareStandardAccessType_ByKeyA_or_KeyB	доступ по ключу A или B

enum XID_eMifareStandardAccessConditionType

Данный перечислимый тип объединяет следующие возможные наборы характеристик доступа:

xidMifareStandardAccessConditionType_0	Тип 0
xidMifareStandardAccessConditionType_1	Тип 1
xidMifareStandardAccessConditionType_2	Тип 2
xidMifareStandardAccessConditionType_3	Тип 3
xidMifareStandardAccessConditionType_4	Тип 4
xidMifareStandardAccessConditionType_5	Тип 5
xidMifareStandardAccessConditionType_6	Тип 6
xidMifareStandardAccessConditionType_	Тип 7

© 2011 MicroEM

7

IXID_MifareStandard_SectorTrailerAccessConditions OfSectorTrailer

Представляет собой указатель на интерфейс типа **IUnknown**, посредством которого можно получить/установить параметры доступа к трейлеру

[in, out] XID_eMifareStandardAccessConditionType AccessConditionType

Задает набор правил доступа

[in, out] XID_eMifareStandardAccessType WriteKeyA

Задает тип доступа для записи ключа A

[in, out] XID_eMifareStandardAccessType ReadKeyA

Задает тип доступа для чтения ключа A

[in, out] XID_eMifareStandardAccessType WriteAccessBits

Задает тип доступа для записи битов доступа

[in, out] XID_eMifareStandardAccessType ReadAccessBits

Задает тип доступа для чтения битов доступа

[in, out] XID_eMifareStandardAccessType WriteKeyB

Задает тип доступа для записи ключа B

[in, out] XID_eMifareStandardAccessType ReadKeyB

Задает тип доступа для чтения ключа B

© 2011 MicroEM

enum XID_eMifareStandardKeyType

Данный перечислимый тип объединяет следующие возможные значения типов ключей:

xidMifareStandardKey_TypeA	ключ A
xidMifareStandardKey_TypeB	ключ B

IXID_Std_MifareULFuncs UltraLight

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.UltraLight**

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками стандарта **Mifare Ultra Light** от **NXP**.

IXID_Std_ISO14443AFuncs ISO14443A

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.UltraLightISO14443A**.

Здесь собраны базовые команды, предназначенные для управления устройствами типа " Считыватель " при работе с метками стандарта **Mifare Ultra Light** от **NXP**.

Подробное описание базовых команд для работы с данным стандартом идентично описанию **IXID_Std_ISO14443AFuncs**

IXID_Cmd_Reader_MifareUL_ReadAllPages ReadAllPages

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.ReadAllPages**

Чтение всех страниц из памяти карты

[out] IXID_MifareUL_Pages out_Pages

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно получить данные о страницах памяти карты

do_ReadAllPages()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareUL_ReadPages ReadPages

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.ReadPages**

Чтение 4х страниц из памяти карты, начиная с указанной

[in, out] byte in_StartPageNumber

Номер страницы (начиная от 0) с которой начать чтение.

Чтение производится 4х страниц сразу, начиная с указанной.

Если **in_StartPageNumber** больше 12, после чтения последней страницы, страницы начнут читаться с 0й.

[out] IXID_MifareUL_Pages out_Pages

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно получить данные о страницах памяти карты

do_ReadPages()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareUL_WritePage WritePage

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.WritePage**

Запись страницы в память карты

[in, out] byte in_PageNumber

Номер страницы (начиная от 0) в которую требуется произвести запись

[out] IXID_MifareUL_Pages in_Pages

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно получить данные о страницах памяти карты

do_WritePage()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareUL_WriteAllPages WriteAllPages

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.WriteAllPages**

Запись всех страниц в память карты

[out] IXID_MifareUL_Pages in_Pages

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно получить данные о страницах памяти карты

do_WriteAllPages()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareUL_ReadLock ReadLock

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.ReadLock**

Чтение страниц карты, хранящих параметры блокировки записи

[out] IXID_MifareUL_Pages out_Pages

© 2011 MicroEM

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно получить данные о страницах памяти карты

do_ReadLock()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareUL_WriteLock WriteLock

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.WriteLock**

Запись страниц карты, хранящих параметры блокировки записи

[out] IXID_MifareUL_Pages in_Pages

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно получить данные о страницах памяти карты

do_WriteLock()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareUL_ReadOTP ReadOTP

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.ReadOTP**

Чтение однократно программируемых бит

[out] IXID_MifareUL_Pages out_Pages

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно получить данные о страницах памяти карты

do_ReadOTP()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareUL_WriteOTP WriteOTP

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.WriteOTP**.

Запись однократно программируемых бит

[out] IXID_MifareUL_Pages in_Pages

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно получить данные о страницах памяти карты

do_WriteOTP()

Запуск выполнения данной команды/функции.

IXID_MifareUL_Pages Pages

Представляет собой указатель на интерфейс типа **IDispatch**, посредством которого можно получить данные о страницах памяти карты

[out] string SerialNumber

Серийный номер метки в виде строки шестнадцатеричных символов

[out] boolean SerialNumberIsOK

true если контрольная сумма серийного номера верная

[out] IXID_MifareUL_LockInfo LockInfo

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно изменять параметры блокировки записи

[in, out] boolean PageLocked

Флаг, указывающий на то, заблокирована ли страница для записи

[in, out] boolean OTP_Locked

Флаг, указывающий на то, заблокированы ли биты **OTP** для записи

[in, out] boolean LockOfOTP_Blocked

Флаг, указывающий на то, возможна ли блокировка **OTP**

[in, out] boolean LockOfPages4to9_Blocked

Флаг, указывающий на то, возможна ли блокировка страниц с 4 по 9

[in, out] boolean LockOfPages10to15_Blocked

Флаг, указывающий на то, возможна ли блокировка страниц с 10 по 15

[out] IXID_MifareUL_OTP OTP

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно изменять биты **OTP**

[in, out] boolean LockBlocked

Флаг, указывающий на то, возможна ли блокировка **OTP**

[in, out] boolean Locked

Флаг, указывающий на то, заблокированы ли биты **OTP** для записи

[in, out] boolean Bits[byte BitIndex]

Бит **OTP**, соответствующий индексу **BitIndex**

[in, out] long Value

Все биты **OTP** в одной переменной

[in, out] byte array PagesArray

Байтовый массив данных страниц

[in, out] string AllPagesAsHexString

Все страницы в виде строки шестнадцатеричных символов

[in, out] string array PagesHexStringArray

Массив данных страниц - в виде строк шестнадцатеричных символов

[out] IXID_MifareUL_Page GetDataPage[byte PageIndex]

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно изменять данные страницы **PageIndex**

[in, out] string DataAsTextString

Данные страницы в виде символьной строки

[in, out] string DataAsHexString

Данные страницы в виде строки шестнадцатеричных символов

© 2011 MicroEM

[in, out] long DataAsValue

Данные страницы в виде переменной типа **long**

[in, out] byte Bytes[byte ByteIndex]

Байт страницы, соответствующий индексу **ByteIndex**

[in, out] boolean Locked

Флаг, указывающий на то, заблокирована ли страница для записи

[out] boolean GetLockBlocked

Флаг, указывающий на то, возможна ли блокировка данной страницы

IXID_Std_MifareDESfire_Funcs DesFire

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire**

Здесь собраны команды, предназначенные для управления устройствами типа "Считыватель " при работе с метками стандарта **Mifare DES Fire** от **NXP**.

IXID_Std_ISO14443AFuncs ISO14443A

Данный интерфейс типа **IDispatch** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.UltraLightISO14443A**.

Здесь собраны базовые команды, предназначенные для управления устройствами типа " Считыватель " при работе с метками стандарта **Mifare DES Fire** от **NXP**.

Подробное описание базовых команд для работы с данным стандартом идентично описанию **IXID_Std_ISO14443AFuncs**

IXID_Cmd_Reader_MifareDESFire_RequestAnswerToSelect RequestAnswerToSelect

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.RequestAnswerToSelect**

Включение режима T=CL.

[in, out] byte out_CardNumber

Номер переведённой в режим T=CL карты.

do_RequestAnswerToSelect()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_Authenticate Authenticate

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Authenticate**

Аутентификация карты/приложения.

[in, out] byte in_KeyNumber

Номер ключа, по которому будет производиться аутентификация.

[in, out] XID_eMifareDESfire_AuthenticationType in_AuthenticationType

Тип аутентификации.

[in, out] string in_KeyAsHexString

Данные ключа в виде строки шестнадцатеричных символов.

do_Authenticate()

Запуск выполнения данной команды/функции.

© 2011 MicroEM

IXID_Cmd_Reader_MifareDESFire_SetConfiguration SetConfiguration

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.SetConfiguration**

Установка конфигурации карты.

[in, out] XID_eMifareDESfire_ConfigurationOptions in_ConfigurationOption

Тип записываемой конфигурации.

[in, out] boolean in_FormatCardDisabled

Отключение возможности форматировать карту.

[in, out] boolean in_RandomIDEnabled

Включение случайного идентификатора.

[in, out] string in_Key

Данные ключа в виде строки шестнадцатеричных символов.

[in, out] byte in_KeyVersion

Версия ключа.

[in, out] string in_ATS

ATS карты в виде строки шестнадцатеричных символов.

do_SetConfiguration()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_FreeMem FreeMem

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.FreeMem**

Освобождение памяти на карте.

[in, out] long out_MemorySize

Размер памяти карты.

do_FreeMem()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_ChangeKeySettings ChangeKeySettings

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.ChangeKeySettings**

Изменение настроек ключа.

[out] IXID_MifareDESFire_PrimaryKeySettings in_PiccMasterKeySettings

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно задать настройки мастер-ключа карты.

[out] IXID_MifareDESFire_ApplicationMasterKeySettings in_ApplicationMasterKeySettings

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно задать настройки мастер-ключа приложения.

do_ChangeKeySettings()

Запуск выполнения данной команды/функции.

© 2011 MicroEM

IXID_Cmd_Reader_MifareDESFire_GetMasterKeySettings GetMasterKeySettings

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.GetMasterKeySettings**

Считывание настроек мастер-ключа карты.

[out] IXID_MifareDESFire_PrimaryKeySettings out_MasterKeySettings

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно прочитать настройки мастер-ключа карты.

[out] IXID_MifareDESFire_KeySettings out_KeySettings

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно прочитать настройки ключа приложения.

do_GetKeySettings()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_ChangeKey ChangeKey

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.ChangeKey**

Смена ключа.

[in, out] byte in_KeyVersion

Версия ключа.

[in, out] string in_Key

Данные ключа в виде строки шестнадцатеричных символов.

[in, out] byte in_KeyNumber

Номер ключа, по которому будет производиться аутентификация.

[in, out] XID_eMifareDESfire_PiccMasterKeyType in_ChangeKeyType

Тип изменяемого ключа.

do_ChangeKey()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_GetKeyVersion GetKeyVersion

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.GetKeyVersion**

Считывание версии ключа.

[in, out] byte in_KeyNumber

Номер ключа, по которому будет производиться аутентификация.

[in, out] byte out_KeyVersion

Версия ключа.

do_GetKeyVersion()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_GetCardID GetCardID

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.GetCardID**

Считывание идентификатора карты (при использовании случайного идентификатора).

© 2011 MicroEM

[in, out] string out_CardUID

Идентификатор метки в виде строки шестнадцатеричных символов.

do_GetCardUID()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_GetDFNames GetDFNames

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.GetDFNames**

Считывание DF-имён файлов.

[out] IXID_MifareDESFire_DFNames out_DFNames

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно прочитать DF-имена файлов.

do_GetDFNames()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_FormatPICC FormatPICC

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.FormatPICC**

Форматирование карты.

do_FormatPICC()

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_GetVersion GetVersion

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.GetVersion**.

Считывание версии карты.

[out] IXID_MifareDESFire_Version out_Version

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно прочитать сведения о версии карты.

do_GetVersion()

Запуск выполнения данной команды/функции.

XID_MifareDESfire_Version - это объект, содержащий информацию о карте, как о продукте, включая версии программного обеспечения нижнего и верхнего уровня на карте.

IXID_Cmd_Reader_MifareDESFire_VersionHardwareInfo HardwareInfo

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.GetVersion.out_Version.HardwareInfo**.

Информация о программном обеспечении нижнего уровня на карте.

[in, out] byte HardwareVendorID

Идентификатор продавца ПО нижнего уровня карты.

[in, out] byte HardwareType

Тип ПО нижнего уровня карты.

[in, out] byte HardwareSubtype

Подтип ПО нижнего уровня карты.

[in, out] byte HardwareMajorVersionNumber

Старший номер версии ПО нижнего уровня карты.

[in, out] byte HardwareMinorVersionNumber

Младший номер версии ПО нижнего уровня карты.

[out] long HardwareStorageSize

Размер памяти для ПО нижнего уровня карты.

[in, out] byte HardwareCommunicationProtocol

Протокол обмена ПО нижнего уровня карты.

IXID_Cmd_Reader_MifareDESFire_VersionSoftwareInfo SoftwareInfo

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.GetVersion.out_Version.SoftwareInfo**.

Информация о программном обеспечении верхнего уровня на карте.

[in, out] byte SoftwareVendorID

Идентификатор продавца ПО верхнего уровня карты.

[in, out] byte SoftwareType

Тип ПО верхнего уровня карты.

© 2011 MicroEM

[in, out] byte SoftwareSubtype

Подтип ПО верхнего уровня карты.

[in, out] byte SoftwareMajorVersionNumber

Старший номер версии ПО верхнего уровня карты.

[in, out] byte SoftwareMinorVersionNumber

Младший номер версии ПО верхнего уровня карты.

[out] long SoftwareStorageSize

Размер памяти для ПО верхнего уровня карты.

[in, out] byte SoftwareCommunicationProtocol

Протокол обмена ПО верхнего уровня карты.

[in, out] string ProductionButchNumber

Номер серии карты в виде строки шестнадцатеричных символов.

[in, out] string SerialNumber

Серийный номер карты в виде строки шестнадцатеричных символов.

[in, out] byte ProductionWeek

© 2011 MicroEM

Байт с закодированной неделей производства карты.

[in, out] byte ProductionYear

Байт с закодированным годом производства карты.

[out] XID_eMifareDESfire_ErrorCodes LastError

Последняя ошибка, возникшая при взаиможействии с картой.

[out] string LastErrorDescription

Строка с описанием последней ошибки, возникшей при взаимодействии с картой.

[out] byte GetNewCardNumber

Возвращает наименьший свободный номер для карты DES Fire, одновременно добавляя его в список занятых номеров.

[out] IXID_MifareDESfire_Applications Applications

Список приложений, формируемый после выполнения команды GetApplicationIDs.

FreeCardNumber
(
[in] byte Number
)

Удаляет из списка занятых номеров карт указанный номер.

XID_MifareDESfire_Applications- это объект-список приложений, формирующийся из считанных командой GetApplicationIDs идентификаторов.

IXID_Cmd_Reader_MifareDESFire_CreateApplication CreateApplication

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications.CreateApplication**

Создание приложения.

[in, out] long in_ApplicationID

Идентификатор приложения

[out] IXID_MifareDESfire_ApplicationMasterKeySettings in_ApplicationMasterKeySettings

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно задать набор установок для мастер-ключа приложения.

[out] IXID_MifareDESfire_ApplicationKeySettings in_ApplicationKeySettings

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно задать набор установок для ключа приложения.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_SelectApplication SelectApplication

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications.SelectApplication**

Выбор приложения.

[in, out] long in_ApplicationID

Идентификатор приложения

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_DeleteApplication DeleteApplication

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications.DeleteApplication**

© 2011 MicroEM

Удаление приложения.

[in, out] long in_ApplicationID

Идентификатор приложения

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_CommitTransaction CommitTransaction

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications.CommitTransaction**

Принятие изменений (копирует данные из резервных буфферов в основные).

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_AbortTransaction AbortTransaction

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications.AbortTransaction**

Отмена изменений (копирует данные из основных буфферов в резервные).

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_GetApplicationIDs GetApplicationIDs

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications.GetApplicationIDs**

Считывает идентификаторы приложений на карте, из которых затем создаётся список XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications .

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_GetKeySettings GetKeySettings

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications.GetKeySettings**

Считывает настройки ключа приложения.

[out] IXID_MifareDESFire_PrimaryKeySettings out_MasterKeySettings

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно прочитать настройки мастер-ключа карты.

[in, out] XID_eMifareDESfire_PiccMasterKeyType out_MasterKeyType

Тип мастер-ключа.

[out] IXID_MifareDESfire_ApplicationMasterKeySettings out_ApplicationMasterKeySettings

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно получить набор установок для мастер-ключа приложения.

[out] IXID_MifareDESfire_ApplicationKeySettings out_ApplicationKeySettings

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно получить набор установок для ключа приложения.

Запуск выполнения данной команды/функции.

[out] long Count

Возвращает текущее количество приложений в списке.

```
[out] IXID_MifareDESfire_Application Item
(
    long Index
)
```

Возвращает указатель на объект приложения с индексом **Index** из внутреннего массива списка приложений (счетчик ссылок на объект данного устройства при этом увеличивается на 1). Это свойство также доступно при указании индекса приложения в скобках сразу же после имени переменной-держателя списка приложений (см. Доступ к свойствам объектов).

© 2011 MicroEM

XID_MifareDESfire_Application-это объект-приложение.

IXID_Cmd_Reader_MifareDESFire_GetApplicationKeySettings GetKeySettings

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].GetKeySettings**

Считывает настройки ключа приложения с индексом **AppIndex**.

[out] IXID_MifareDESfire_ApplicationMasterKeySettings out_ApplicationMasterKeySettings

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно получить набор установок для мастер-ключа приложения.

[out] IXID_MifareDESfire_ApplicationKeySettings out_ApplicationKeySettings

Возвращает указатель на интерфейс типа **IUnknown**, посредством которого можно получить набор установок для ключа приложения.

Enter topic text here.

[in, out] long ID

Идентификатор приложения.

[out] IXID_MifareDESfire_Files Files

Список файлов, формируемый после выполнения команды GetFileIDs.

Удаляет данное приложение из карты.

Выбирает данное приложение для дальнейшей работы.

Принятие изменений для данного приложения (копирует данные из резервных буфферов в основные).

Отмена изменений для данного приложения (копирует данные из основных буфферов в резервные).

XID_MifareDESfire_Files это объект-список файлов, формирующийся из считанных командой GetFileIDs идентификаторов.

IXID_Cmd_Reader_MifareDESFire_GetFileIDs GetFileIDs

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.GetFileIDs**

Считывает идентификаторы файлов приложения с индексом **AppIndex** и формирует из них объект-список файлов XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_GetISOFileIDs GetISOFileIDs

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.GetISOFileIDs**

Считывает ISO-идентификаторы файлов приложения с индексом **AppIndex**.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_GetFileSettings GetFileSettings

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.GetFileSettings**

Считывает настройки файлов для приложения с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

[out] XID_eMifareDESfire_CodingOfFileTypes out_FileType

Тип файла.

[out] XID_eMifareDESfire_CodingOfCommunicationSettings out_CommunicationSettings

Настройки канала обмена.

[out] XID_eMifareDESfire_CodingOfAccessRights out_ReadAccess

Права доступа на чтение.

[out] XID_eMifareDESfire_CodingOfAccessRights out_WriteAccess

Права доступа на запись.

[out] XID_eMifareDESfire_CodingOfAccessRights out_ReadAndWriteAccess

Права доступа на запись и чтение.

[out] XID_eMifareDESfire_CodingOfAccessRights out_ChangeAccessRights

Права доступа на смену прав доступа.

[out] long out_FileSize

Размер файла данных (стандартного или с резервным копированием).

[out] long out_LowerLimit

Нижний предел значения для файла значений.

[out] long out_UpperLimit

Верхний предел значения для файла значений.

[out] long out_LimitedCreditValue

Предел для приращения значения функцией LimitedCredit.

[out] boolean out_LimitedCreditEnabled

Включение функции LimitedCredit.

[out] long out_RecordSize

Размер записи для файла записей (линецного или циклического).

[out] long out_MaximumNumberOfRecords

Максимальное количество записей в файле записей (линецном или циклическом).

[out] long out_CurrentNumberOfRecords

Текущее количество записей в файле записей (линецном или циклическом).

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_ChangeFileSettings ChangeFileSettings

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.ChangeFileSettings**

Меняет настройки файлов для приложения с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings in_CommunicationSettings

Настройки канала обмена.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ReadAccess

Права доступа на чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_WriteAccess

Права доступа на запись.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ReadAndWriteAccess

Права доступа на запись и чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ChangeAccessRights

Права доступа на смену прав доступа.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_CreateStdDataFile CreateStdDataFile

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.CreateStdDataFile**

Создаёт файл данных для приложения с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings in_CommunicationSettings

Настройки канала обмена.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ReadAccess

Права доступа на чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_WriteAccess

Права доступа на запись.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ReadAndWriteAccess

Права доступа на запись и чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ChangeAccessRights

Права доступа на смену прав доступа.

[in, out] long in_FileSize

Размер файла данных (стандартного или с резервным копированием).

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_CreateBackupDataFile CreateBackupDataFile

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.CreateBackupDataFile**

Создаёт файл данных с резервным копированием для приложения с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings in_CommunicationSettings

Настройки канала обмена.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ReadAccess

Права доступа на чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_WriteAccess

© 2011 MicroEM

Права доступа на запись.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ReadAndWriteAccess

Права доступа на запись и чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ChangeAccessRights

Права доступа на смену прав доступа.

[in, out] long in_FileSize

Размер файла данных (стандартного или с резервным копированием).

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_CreateValueFile CreateValueFile

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.CreateValueFile**

Создаёт файл-значение с резервным копированием для приложения с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings in_CommunicationSettings

Настройки канала обмена.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ReadAccess

Права доступа на чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_WriteAccess

Права доступа на запись.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ReadAndWriteAccess

Права доступа на запись и чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ChangeAccessRights

Права доступа на смену прав доступа.

[in, out] long in_LowerLimit

Нижний предел значения для файла значений.

[in, out] long in_UpperLimit

Верхний предел значения для файла значений.

[in, out] long in_Value

Значение, участвующее в операции для файла значения.

[in, out] boolean in_LimitedCreditEnabled

Включение функции LimitedCredit.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_CreateLinearRecordFile CreateLinearRecordFile

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.CreateLinearRecordFile**

Создаёт линейный файл записей с резервным копированием для приложения с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings in_CommunicationSettings

Настройки канала обмена.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ReadAccess

Права доступа на чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_WriteAccess

Права доступа на запись.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ReadAndWriteAccess

Права доступа на запись и чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ChangeAccessRights

Права доступа на смену прав доступа.

[in, out] long in_RecordSize

Размер записи для файла записей (линецного или циклического).

[in, out] long in_MaximumNumberOfRecords

Максимальное количество записей в файле записей (линецном или циклическом).

© 2011 MicroEM

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_CreateCyclicRecordFile CreateCyclicRecordFile

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.CreateCyclicRecordFile**

Создаёт циклический файл записей с резервным копированием для приложения с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings in_CommunicationSettings

Настройки канала обмена.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ReadAccess

Права доступа на чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_WriteAccess

Права доступа на запись.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ReadAndWriteAccess

Права доступа на запись и чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ChangeAccessRights

Права доступа на смену прав доступа.

[in, out] long in_RecordSize

© 2011 MicroEM

Размер записи для файла записей (линецного или циклического).

[in, out] long in_MaximumNumberOfRecords

Максимальное количество записей в файле записей (линецном или циклическом).

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_DeleteFile DeleteFile

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.DeleteFile**

Удаляет файл в приложении с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_ReadData ReadData

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.ReadData**

Считывает данные из файла данных (стандартного или с резервным копированием) в приложении с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings in_CommunicationSettings

Настройки канала обмена.

[in, out] long in_Offset

Позиция байта (начальная запись для файла записей), с которого следует начинать чтение данных из файла данных (записей).

[in, out] long in_Length

Количество байт (записей для файла записей) которое следует прочитать из файла данных (записей).

[in, out] string out_DataAsHexString

Прочитанные данные в виде строки шестнадцатеричных символов.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_WriteData WriteData

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.WriteData**

Записывает данные в файл данных (стандартный или с резервным копированием) в приложении с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings in_CommunicationSettings

Настройки канала обмена.

[in, out] long in_Offset

Позиция байта (начальная запись для файла записей), с которого следует начинать чтение данных из файла данных (записей).

© 2011 MicroEM

[in, out] long in_Length

Количество байт (записей для файла записей) которое следует прочитать из файла данных (записей).

[in, out] string in_DataAsHexString

Данные для записи в виде строки шестнадцатеричных символов.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_GetValue GetValue

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.GetValue**

Считывает значение из файла-значения в приложении с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings in_CommunicationSettings

Настройки канала обмена.

[in, out] long out_Value

Прочитанное значение.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_Credit Credit

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.Credit**

Увеличивает на заданную величину значение в файле-значении в приложении с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings in_CommunicationSettings

Настройки канала обмена.

[in, out] long in_Value

Значение, участвующее в операции для файла значения.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_Debit Debit

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.Debit**

Уменьшает на заданную величину значение в файле-значении в приложении с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings in_CommunicationSettings

Настройки канала обмена.

[in, out] long in_Value

© 2011 MicroEM

Значение, участвующее в операции для файла значения.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_LimitedCredit LimitedCredit

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.LimitedCredit**

Увеличивает с ограничением на заданную величину значение в файле-значении в приложении с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings in_CommunicationSettings

Настройки канала обмена.

[in, out] long in_Value

Значение, участвующее в операции для файла значения.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_WriteRecord WriteRecord

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.WriteRecord**

Добавляет запись в файл записей (линейный или циклический) в приложении с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings in_CommunicationSettings

Настройки канала обмена.

[in, out] long in_Offset

Позиция байта (начальная запись для файла записей), с которого следует начинать чтение данных из файла данных (записей).

[in, out] long in_Length

Количество байт (записей для файла записей) которое следует прочитать из файла данных (записей).

[in, out] string in_DataAsHexString

Данные для записи в виде строки шестнадцатеричных символов.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_ClearRecordFile ClearRecordFile

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.ClearRecordFile**

Очищает файл записей в приложении с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_ReadRecords ReadRecords

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.**

TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files.ReadRecords

Считывает записи из файла записей (линейного или циклического) в приложении с индексом **AppIndex**.

[in, out] byte in_FileID

Идентификатор файла.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings in_CommunicationSettings

Настройки канала обмена.

[in, out] long in_Offset

Позиция байта (начальная запись для файла записей), с которого следует начинать чтение данных из файла данных (записей).

[in, out] long in_Length

Количество байт (записей для файла записей) которое следует прочитать из файла данных (записей).

[in, out] string out_DataAsHexString

Прочитанные данные в виде строки шестнадцатеричных символов.

Запуск выполнения данной команды/функции.

[out] long Count

Возвращает текущее количество файлов в приложении.

[out] IXID_MifareDESfire_File Item
(
long Index
)

Возвращает указатель на объект файла с индексом **Index** из внутреннего массива списка файлов (счетчик ссылок на объект данного устройства при этом увеличивается на 1). Это свойство также доступно при указании индекса файла в скобках сразу же после имени переменной-держателя списка файлов (см. Доступ к свойствам объектов).

enum XID_eMifareDESfire_CodingOfFileTypes

Данный перечислимый тип объединяет возможные значения типов файлов:

xidMifareDESfire_FileType_StandardDataFile	стандартный файл данных
xidMifareDESfire_FileType_BackupDataFile	файл данных с резервным копированием
xidMifareDESfire_FileType_ValueFileWithBackup	файл-значение с резервным копированием
xidMifareDESfire_FileType_LinearRecordFileWithBackup	линейный файл записей с резервным копированием
xidMifareDESfire_FileType_CyclicRecordFileWithBackup	циклический файл записей с резервным копированием

enum XID_eMifareDESfire_CodingOfCommunicationSettings

Данный перечислимый тип объединяет возможные значения режимов обмена данными с картой:

xidMifareDESfire_CommMode_PlainCommunication	открытый режим передачи
xidMifareDESfire_CommMode_PlainCommunicationWithDesOr3DesMAC	режим передачи с MAC-подписью
xidMifareDESfire_CommMode_FullyDesOr3DesEncipheredCommunication	полностью зашифрованный обмен данными

enum XID_eMifareDESfire_CodingOfAccessRights

Данный перечислимый тип объединяет возможные типы доступа к файлам:

xidMifareDESfire_AccessRight_ByKey0	доступ по ключу 0
xidMifareDESfire_AccessRight_ByKey1	доступ по ключу 1
xidMifareDESfire_AccessRight_ByKey2	доступ по ключу 2
xidMifareDESfire_AccessRight_ByKey3	доступ по ключу 3
xidMifareDESfire_AccessRight_ByKey4	доступ по ключу 4
xidMifareDESfire_AccessRight_ByKey5	доступ по ключу 5
xidMifareDESfire_AccessRight_ByKey6	доступ по ключу 6
xidMifareDESfire_AccessRight_ByKey7	доступ по ключу 7
xidMifareDESfire_AccessRight_ByKey8	доступ по ключу 8
xidMifareDESfire_AccessRight_ByKey9	доступ по ключу 9

© 2011 MicroEM

xidMifareDESfire_AccessRight_ByKey10	доступ по ключу 10
xidMifareDESfire_AccessRight_ByKey11	доступ по ключу 11
xidMifareDESfire_AccessRight_ByKey12	доступ по ключу 12
xidMifareDESfire_AccessRight_ByKey13	доступ по ключу 13
xidMifareDESfire_AccessRight_FreeAccess	Открытый доступ
xidMifareDESfire_AccessRight_DenyAccess	Доступ закрыт

XID_MifareDESfire_File - это объект-файл из приложения.

IXID_Cmd_Reader_MifareDESFire_FileGetSettings GetSettings

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files[FileIndex].GetSettings**

Считывает настройки для файла с индексом **FileIndex**приложения с индексом **AppIndex**.

[out] XID_eMifareDESfire_CodingOfFileTypes out_FileType

Тип файла.

[out] XID_eMifareDESfire_CodingOfCommunicationSettings out_CommunicationSettings

Настройки канала обмена.

[out] XID_eMifareDESfire_CodingOfAccessRights out_ReadAccess

Права доступа на чтение.

[out] XID_eMifareDESfire_CodingOfAccessRights out_WriteAccess

Права доступа на запись.

[out] XID_eMifareDESfire_CodingOfAccessRights out_ReadAndWriteAccess

Права доступа на запись и чтение.

[out] XID_eMifareDESfire_CodingOfAccessRights out_ChangeAccessRights

Права доступа на смену прав доступа.

[out] long out_FileSize

Размер файла данных (стандартного или с резервным копированием).

[out] long out_LowerLimit

Нижний предел значения для файла значений.

[out] long out_UpperLimit

Верхний предел значения для файла значений.

[out] long out_LimitedCreditValue

Предел для приращения значения функцией LimitedCredit.

[out] boolean out_LimitedCreditEnabled

Включение функции LimitedCredit.

[out] long out_RecordSize

Размер записи для файла записей (линецного или циклического).

[out] long out_MaximumNumberOfRecords

Максимальное количество записей в файле записей (линецном или циклическом).

[out] long out_CurrentNumberOfRecords

Текущее количество записей в файле записей (линецном или циклическом).

Запуск выполнения данной команды/функции.

© 2011 MicroEM

IXID_Cmd_Reader_MifareDESFire_FileChangeSettings ChangeSettings

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files[FileIndex].ChangeSettings**

Меняет настройки для файла с индексом **FileIndex**приложения с индексом **AppIndex**.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings in_CommunicationSettings

Настройки канала обмена.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ReadAccess

Права доступа на чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_WriteAccess

Права доступа на запись.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ReadAndWriteAccess

Права доступа на запись и чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights in_ChangeAccessRights

Права доступа на смену прав доступа.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_FileReadData ReadData

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files[FileIndex].ReadData**

Считывает данные из файла данных (стандартного или с резервным копированием) с индексом **FileIndex**приложения с индексом **AppIndex**.

[in, out] long in_Offset

Позиция байта (начальная запись для файла записей), с которого следует начинать чтение данных из файла данных (записей).

[in, out] long in_Length

Количество байт (записей для файла записей) которое следует прочитать из файла данных (записей).

[in, out] string out_DataAsHexString

Прочитанные данные в виде строки шестнадцатеричных символов.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_FileWriteData WriteData

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files[FileIndex].WriteData**

Записывает данные в файл данных (стандартный или с резервным копированием) с индексом **FileIndex**приложения с индексом **AppIndex**.

[in, out] long in_Offset

Позиция байта (начальная запись для файла записей), с которого следует начинать чтение данных из файла данных (записей).

[in, out] long in_Length

Количество байт (записей для файла записей) которое следует прочитать из файла данных (записей).

[in, out] string in_DataAsHexString

Данные для записи в виде строки шестнадцатеричных символов.

© 2011 MicroEM

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_FileGetValue GetValue

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files[FileIndex].GetValue**

Считывает значение из файла значений с индексом **FileIndex**приложения с индексом **AppIndex**.

[in, out] long out_Value

Прочитанное значение.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_FileCredit Credit

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files[FileIndex].Credit**

Увеличивает на заданную величину значение из файла значений с индексом **FileIndex** приложения с индексом **AppIndex**.

[in, out] long in_Value

Значение, участвующее в операции для файла значения.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_FileDebit Debit

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files[FileIndex].Debit**

Уменьшает на заданную величину значение из файла значений с индексом **FileIndex** приложения с индексом **AppIndex**.

[in, out] long in_Value

Значение, участвующее в операции для файла значения.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_FileLimitedCredit LimitedCredit

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files[FileIndex].LimitedCredit**

Увеличивает ограниченно на заданную величину значение из файла значений с индексом **FileIndex**приложения с индексом **AppIndex**.

[in, out] long in_Value

Значение, участвующее в операции для файла значения.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_FileWriteRecord WriteRecord

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader.TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files[FileIndex].WriteRecord**

Добавляет запись в файл записей (линейный или циклический) с индексом **FileIndex** приложения с индексом **AppIndex**.

[in, out] long in_Offset

Позиция байта (начальная запись для файла записей), с которого следует начинать чтение данных из файла данных (записей).

[in, out] long in_Length

Количество байт (записей для файла записей) которое следует прочитать из файла данных (записей).

[in, out] string in_DataAsHexString

Данные для записи в виде строки шестнадцатеричных символов.

© 2011 MicroEM

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_FileClearRecordFile ClearRecordFile

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files[FileIndex].ClearRecordFile**

Очищает файл записей (линейный или циклический) с индексом **FileIndex**приложения с индексом **AppIndex**.

Запуск выполнения данной команды/функции.

IXID_Cmd_Reader_MifareDESFire_FileReadRecords ReadRecords

Данный интерфейс типа **IUnknown** доступен через **XID_ControlX.QueryDevices.Reader. TagStandards.Philips.Mifare.DesFire.Applications[AppIndex].Files[FileIndex].ReadRecords**

Считывает записи из файла записей (линейный или циклический) с индексом **FileIndex** приложения с индексом **AppIndex**.

[in, out] long in_Offset

Позиция байта (начальная запись для файла записей), с которого следует начинать чтение данных из файла данных (записей).

[in, out] long in_Length

Количество байт (записей для файла записей) которое следует прочитать из файла данных (записей).

[in, out] string out_DataAsHexString

Прочитанные данные в виде строки шестнадцатеричных символов.

Запуск выполнения данной команды/функции.

[in, out] byte ID

Идентификатор файла.

[in, out] word ISOID

ISO-идентификатор файла.

[in, out] XID_eMifareDESfire_CodingOfFileTypes FileType

Тип файла.

[in, out] XID_eMifareDESfire_CodingOfCommunicationSettings CommunicationSettings

Настройки канала обмена.

[in, out] XID_eMifareDESfire_CodingOfAccessRights ReadAccess

Права доступа на чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights WriteAccess

Права доступа на запись.

[in, out] XID_eMifareDESfire_CodingOfAccessRights ReadAndWriteAccess

Права доступа на запись и чтение.

[in, out] XID_eMifareDESfire_CodingOfAccessRights ChangeAccessRights

Права доступа на смену прав доступа.

Удаляет данный файл из текущего приложения.

XID_MifareDESfire_ApplicationKeySettings- это объект, содержащий настройки для ключа приложения.

[in, out] long MaximumNumberOfKeys

Максимальное количество ключей в приложении.

[in, out] XID_eMifareDESfire_UsageOf2ByteFileIdentifiers UsageOf2ByteFileIdentifiers

Использование 2-байтных идентификаторов файлов.

[in, out] XID_eMifareDESfire_ApplicationCryptoMethod ApplicationCryptoMethod

Тип шифрования в транзакциях приложения.

enum XID_eMifareDESfire_UsageOf2ByteFileIdentifiers

Данный перечислимый тип объединяет возможные варианты использования 2-байтовых идентификаторов файлов:

xidMifareDESfire_NotSupported2ByteFileIdentifiers	2-байтные идентификаторы не поддерживаются
xidMifareDESfire_Supported2ByteFileIdentifiers	2-байтные идентификаторы поддерживаются

enum XID_eMifareDESfire_ApplicationCryptoMethod

Данный перечислимый тип объединяет возможные варианты шифрования транзакций:

xidMifareDESfire_DES_and_3DES_OperationForTheWholeApplication	Операции DES и 3DES для всего приложения
xidMifareDESfire_3K3DES_OperationForTheWholeApplication	Операции 3K3DES для всего приложения
xidMifareDESfire_AES_OperationForTheWholeApplication	Операции AES для всего приложения

XID_MifareDESfire_ApplicationMasterKeySettings- это объект, содержащий настройки для мастер-ключа приложения.

[in, out] XID_eMifareDESfire_AccessRightsForChangingApplicationKeys AccessRightsForChangingApplicationKeys

Доступ к изменению ключей приложения.

[in, out] XID_eMifareDESfire_AccessToApplicationMasterKeySettingsChanging

AccessToApplicationMasterKeySettingsChanging

Доступ к изменению настроек мастер-ключа приложения.

[in, out] XID_eMifareDESfire_CreateAndDeleteFileAccessByApplicationMasterKey CreateAndDeleteFileAccessByApplicationMasterKey

Возможность создавать и удалять файлы по мастер-ключу приложения.

[in, out] XID_eMifareDESfire_FileDirectoryAccessByApplicationMasterKey FileDirectoryAccessByApplicationMasterKey

Доступ к списку файлов по мастер-ключу приложения.

[in, out] XID_eMifareDESfire_ApplicationMasterKeyChangeability ApplicationMasterKeyChangeability

Возможность изменить мастер-ключ приложения.

enum XID_eMifareDESfire_AccessRightsForChangingApplicationKeys

Данный перечислимый тип объединяет возможные варианты доступа для смены ключей приложения:

xidMifareDESfire_AuthenticationWithApplicat onMasterKeyNecessaryToChangeAnyKey	для смены любого ключа приложения требуется аутентификация по мастер-ключу приложения
xidMifareDESfire_AuthenticationWithKey1Nec essaryToChangeAnyKey	необходима аутентификация по ключу 1
xidMifareDESfire_AuthenticationWithKey2Nec essaryToChangeAnyKey	необходима аутентификация по ключу 2
xidMifareDESfire_AuthenticationWithKey3Nec essaryToChangeAnyKey	необходима аутентификация по ключу 3
xidMifareDESfire_AuthenticationWithKey4Nec essaryToChangeAnyKey	необходима аутентификация по ключу 4
xidMifareDESfire_AuthenticationWithKey5Nec essaryToChangeAnyKey	необходима аутентификация по ключу 5
xidMifareDESfire_AuthenticationWithKey6Nec essaryToChangeAnyKey	необходима аутентификация по ключу 6
xidMifareDESfire_AuthenticationWithKey7Nec essaryToChangeAnyKey	необходима аутентификация по ключу 7
xidMifareDESfire_AuthenticationWithKey8Nec essaryToChangeAnyKey	необходима аутентификация по ключу 8

© 2011 MicroEM

xidMifareDESfire_AuthenticationWithKey9Necessary ToChangeAnyKey	необходима аутентификация по ключу 9
xidMifareDESfire_AuthenticationWithKey10Necessary ToChangeAnyKey	необходима аутентификация по ключу 10
xidMifareDESfire_AuthenticationWithKey11Necessary ToChangeAnyKey	необходима аутентификация по ключу 11
xidMifareDESfire_AuthenticationWithKey12Necessary ToChangeAnyKey	необходима аутентификация по ключу 12
xidMifareDESfire_AuthenticationWithKey13Necessary ToChangeAnyKey	необходима аутентификация по ключу 13
xidMifareDESfire_AuthenticationWithTheKeyToBeChangedNecessary ToChangeAnyKey	для смены любого ключа приложения требуется аутентификация по сменяемому ключу приложения
xidMifareDESfire_AllKeysWithinApplicationAreFrozen	все ключи в приложении заморожены

enum XID_eMifareDESfire_AccessToApplicationMasterKeySettingsChanging

Данный перечислимый тип объединяет возможные варианты доступа для смены настроек мастер-ключа приложения:

xidMifareDESfire_ConfigurationNotChangeableAnymore	изменение настроек мастер-ключа карты больше не возможно
xidMifareDESfire_ChangeableByApplicationMasterKeyAuthentication	возможно изменение настроек мастер-ключа приложения после аутентификации мастер-ключём приложения

enum XID_eMifareDESfire_CreateAndDeleteFileAccessByApplicationMasterKey

Данный перечислимый тип объединяет возможные варианты доступа для создания и удаления файлов:

xidMifareDESfire_ApplicationMasterKeyAuthenticationNeeded	требуется утентификация по мастер-ключу приложения
xidMifareDESfire_NoApplicationMasterKeyAuthenticationNeeded	аутентификации по мастер-ключу приложения не требуется

enum XID_eMifareDESfire_FileDirectoryAccessByApplicationMasterKey

Данный перечислимый тип объединяет возможные варианты доступа к списку фийлов в приложении на карте:

xidMifareDESfire_ApplicationMasterKeyAuthenticationIsRequired	требуется аутентификация по мастер-ключу приложения
xidMifareDESfire_NoApplicationMasterKeyAuthenticationRequired	аутентификация по мастер-ключу приложения не требуется

enum XID_eMifareDESfire_ApplicationMasterKeyChangeability

Данный перечислимый тип объединяет возможные варианты доступа для смены мастер-ключа приложения:

xidMifareDESfire_ApplicationMasterKeyIsNot Changeable	мастер-ключ приложенияизменить нельзя
xidMifareDESfire_ApplicationMasterKeyIsChangeable	допускается смена мастер-ключа приложения

enum XID_eMifareDESfire_PiccMasterKeyType

Данный перечислимый тип объединяет возможные значения типов мастер-ключей карты:

xidMifareDESfire_PiccMasterKeyType_DES_or_3DES	ключ типа DES или 3DES
xidMifareDESfire_PiccMasterKeyType_3K3DES	ключ типа 3K3DES
xidMifareDESfire_PiccMasterKeyType_AES	ключ типа AES

XID_MifareDESfire_DFNames - это объект-список DF-имён файлов, формирующийся из считанных командой GetDFNames значений.

[out] long Count

Возвращает текущее количество считанных DF-имён.

[out] IXID_MifareDESfire_DFName Item
(
long Index
)

Возвращает указатель на объект DF-имени файла с индексом **Index** из внутреннего массива списка DF-имён файлов (счетчик ссылок на объект при этом увеличивается на 1). Это свойство также доступно при указании индекса DF-имени файла в скобках сразу же после имени переменной-держателя списка DF-имён файлов (см. Доступ к свойствам объектов).

XID_MifareDESfire_DFName - это объект, содержащий параметры считанного DF-имени файла.

[in, out] long AID

Идентификатор приложения.

[in, out] short FID

Идентификатор файла.

[in, out] string DFName

Строка, содержащая DF-имя файла.

XID_MifareDESfire_KeySettings - это объект, содержащий настройки для ключа карты.

[in, out] long MaximumNumberOfKeys

Максимальное количество ключей в приложении.

[in, out] XID_eMifareDESfire_UsageOf2ByteFileIdentifiers UsageOf2ByteFileIdentifiers

Использование 2-байтных идентификаторов файлов.

[in, out] XID_eMifareDESfire_ApplicationCryptoMethod ApplicationCryptoMethod

Тип шифрования в транзакциях приложения.

enum XID_eMifareDESfire_UsageOf2ByteFileIdentifiers

Данный перечислимый тип объединяет возможные варианты использования 2-байтовых идентификаторов файлов:

xidMifareDESfire_NotSupported2ByteFileIdentifiers	2-байтные идентификаторы не поддерживаются
xidMifareDESfire_Supported2ByteFileIdentifiers	2-байтные идентификаторы поддерживаются

enum XID_eMifareDESfire_ApplicationCryptoMethod

Данный перечислимый тип объединяет возможные варианты шифрования транзакций:

xidMifareDESfire_DES_and_3DES_Operatic ForTheWholeApplication	Операции DES и 3DES для всего приложения
xidMifareDESfire_3K3DES_OperationForThe WholeApplication	Операции 3K3DES для всего приложения
xidMifareDESfire_AES_OperationForTheWh leApplication	Операции AES для всего приложения

XID_MifareDESfire_PrimaryKeySettings- это объект, содержащий настройки для мастер-ключа карты.

[in, out] XID_eMifareDESfire_AccessToMasterKeySettingsChanging AccessToMasterKeySettingsChanging

Доступ к изменению настроек мастер-ключа карты.

[in, out] XID_eMifareDESfire_CreateAndDeleteApplicationAccessByMasterKey CreateAndDeleteApplicationAccessByMasterKey

Возможность создать и удалить приложение по мастер-ключу карты.

[in, out] XID_eMifareDESfire_ApplicationDirectoryAccessByMasterKey ApplicationDirectoryAccessByMasterKey

Доступ к списку приложений по мастер-ключу карты.

[in, out] XID_eMifareDESfire_MasterKeyChangeability MasterKeyChangeability

Возможность изменить мастер-ключ карты.

enum XID_eMifareDESfire_AccessToMasterKeySettingsChanging

Данный перечислимый тип объединяет возможные варианты доступа для смены настроек мастер-ключа карты:

xidMifareDESfire_ConfigurationNotChangea bleAnymore	изменение настроек мастер-ключа карты больше не возможно
xidMifareDESfire_ChangeableByMasterKeyA uthentication	возможно изменение настроек мастер-ключа карты после аутентификации мастер-ключём карты

enum XID_eMifareDESfire_CreateAndDeleteApplicationAccessByMasterKey

Данный перечислимый тип объединяет возможные варианты доступа для создания и удаления файлов:

xidMifareDESfire_MasterKeyAuthenticationNeeded	требуется утентификация по мастер-ключу карты
xidMifareDESfire_NoMasterKeyAuthenticationNeeded	аутентификации по мастер-ключу карты не требуется

enum XID_eMifareDESfire_ApplicationDirectoryAccessByMasterKey

Данный перечислимый тип объединяет возможные варианты доступа к списку приложений на карте:

xidMifareDESfire_MasterKeyAuthenticationIsRequired	требуется аутентификация по мастер-ключу карты
xidMifareDESfire_NoMasterKeyAuthenticationRequired	аутентификация по мастер-ключу карты не требуется

enum XID_eMifareDESfire_MasterKeyChangeability

Данный перечислимый тип объединяет возможные варианты доступа для смены мастер-ключа карты:

xidMifareDESfire_MasterKeyIsNotChangeable	мастер-ключ карты изменить нельзя
xidMifareDESfire_MasterKeyIsChangeable	допускается смена мастер-ключа карты

enum XID_eMifareDESfire_ErrorCodes

Данный перечислимый тип объединяет возможные значения возвращаемых картой кодов ошибок:

xidMifareDESfire_OperationOK	операция прошла успешно
xidMifareDESfire_NoChanges	нет изменений
xidMifareDESfire_OutOfEepromError	не хватило памяти
xidMifareDESfire_IllegalCommandCode	неверный код комманды
xidMifareDESfire_IntegrityError	ошибка целостности данных
xidMifareDESfire_NoSuchKey	нет такого ключа
xidMifareDESfire_LengthError	неверная длина данных
xidMifareDESfire_PermissionDenied	доступ запрещён
xidMifareDESfire_ParameterError	ошибка в параметре
xidMifareDESfire_ApplicationNotFound	приложение не найдено
xidMifareDESfire_ApplicationIntegrityError	ошибка целостности приложения
xidMifareDESfire_AuthenticationError	ошибка аутентификации

© 2011 MicroEM

xidMifareDESfire_AdditionalFrame	требуется дополнительный кадр
xidMifareDESfire_BoundaryError	не верный диапазон
xidMifareDESfire_PiccIntegrityError	ошибка целостности карты
xidMifareDESfire_CommandAborted	комманда отменена
xidMifareDESfire_PiccDisabledError	карта отключена
xidMifareDESfire_CountError	ошибка счёта
xidMifareDESfire_DuplicateError	ошибка - дублирование
xidMifareDESfire_EepromError	ошибка в памяти карты
xidMifareDESfire_FileNotFound	файл не найден
xidMifareDESfire_FileIntegrityError	ошибка целостности файла

enum XID_eMifareDESfire_AuthenticationType

Данный перечислимый тип объединяет возможные значения типов аутентификации:

xidMifareDESfire_AuthenticationDES	DES
xidMifareDESfire_Authentication3DES	3DES
xidMifareDESfire_Authentication3K3DES	3K3DES
xidMifareDESfire_AuthenticationAES	AES

enum XID_eMifareDESfire_ConfigurationOptions

Данный перечислимый тип объединяет возможные значения параметров конфигурации:

xidMifareDESfire_ConfigurationOption_ConfigByte	параметр - байт конфигурации
xidMifareDESfire_ConfigurationOption_KeyData	параметры - ключ и версия ключа
xidMifareDESfire_ConfigurationOption_ATS	параметр - ATS карты

5.1.5 XID_TagsList

XID_TagsList является одним из базовых объектов библиотеки и представляет собой список меток с набором свойств и методов, необходимых для добавления, удаления новых элементов и пр..

5.1.5.1 Методы

5.1.5.1.1 Clear

Clear()

Очищает список меток.

5.1.5.1.2 Add

```
Add
(
    XID_Tag tag
)
```

Добавляет метку **tag** в список.

Объект **tag** должен быть предварительно создан.

5.1.5.1.3 AddTagsList

```
AddTagsList
(
    XID_TagsList tags
)
```

Добавляет список меток **tags** в список.

Объект **tags** должен быть предварительно создан.

5.1.5.1.4 SubstractTagsList

```
SubstractTagsList
(
    XID_TagsList tags
)
```

Удаляет из списка меток список **tags**.

Объект **tags** должен быть предварительно создан.

5.1.5.1.5 CopyTo

```
CopyTo
(
    XID_TagsList tags
)
```

Копирует содержимое текущего списка меток в **tags**.

Объект **tags** должен быть предварительно создан.

5.1.5.1.6 Remove

```
Remove
(
    XID_Tag tag
)
```

Удаляет из списка метку, идентичную метке **tag**.

Объект **tag** должен быть предварительно создан.

5.1.5.1.7 DeleteTagsWithID

```
DeleteTagsWithID
(
    string tagID
)
```

Удаляет из списка метки и идентификатором **tagID**.

5.1.5.1.8 GetTagsByType

```
GetTagsByType
(
    XID_TagsList tags,
    XID_eTagType tagtype
)
```

Записывает из текущего списка меток в **tags** метки, относящиеся к типу **tagtype**.

Объект **tags** должен быть предварительно создан.

5.1.5.1.9 MutualExcludeDuplicates

```
MutualExcludeDuplicates
(
    XID_TagsList tags,
    boolean CompareData
)
```

Удаляет из текущего списка и из списка меток **tags** те метки, которые присутствуют в обоих списках (при истинном значении **CompareData,** сравниваются также данные меток).

Объект **tags** должен быть предварительно создан.

© 2011 MicroEM

5.1.5.1.10 SortByType

SortByType()

Производит сортировку списка меток по их типам.

5.1.5.1.11 SetTextStringToAllTags

SetTextStringToAllTags
(
[in] string TextString
)

Записывает во все метки в списке символьную строку **TextString**.

5.1.5.1.12 SetHexStringToAllTags

SetHexStringToAllTags
(
[in] string HexString
)

Записывает во все метки в списке данные в виде строки шестнадцатеричных символов **HexString**.

5.1.5.2 Свойства

5.1.5.2.1 Count

[out] long Count

Возвращает текущее количество меток в списке.

5.1.5.2.2 Item

[out] XID_Tags Item
(
long Index
)

Возвращает указатель на объект метки с индексом **Index** из внутреннего массива списка меток (счетчик ссылок на объект данного устройства при этом увеличивается на 1). Это свойство также доступно при указании индекса метки в скобках сразу же после имени переменной-держателя

списка меток (см. Доступ к свойствам объектов).

5.1.5.2.3 GetTagIndex

[out] long GetTagIndex
(
XID_Tag tag
)

Возвращает индекс метки из списка меток, идентичного метке **tag**.

Если такая метка отсутствует в списке, возвращается "-1".

Объект **tag** должен быть предварительно создан.

5.1.6 XID_Tag

XID_Tag является одним из базовых объектов библиотеки и включает в себя набор общих свойств такого элемента системы идентификации, как метка.

5.1.6.1 Методы

5.1.6.1.1 CloneFrom

CloneFrom
(
XID_Tag tag
)

Устанавливает в текущей метке свойства, идентичные свойствам **tag**.

Объект **tag** должен быть предварительно создан.

5.1.6.2 Свойства

5.1.6.2.1 ID

[in, out] string ID

Идентификатор метки в шестнадцатеричном виде.

5.1.6.2.2 Type

[in, out] XID_eTagType Type

Тип метки.

5.1.6.2.3 TypeStr

[in, out] string TypeStr

Тип метки в виде символьной строки.

5.1.6.2.4 Vendor

[in, out] string Vendor

Наименование производителя метки в виде строки.

5.1.6.2.5 LastOperationStatus

[in, out] XID_eSlotErrorCodes LastOperationStatus

Статус выполнения последней операции над меткой.

5.1.6.2.6 BlockSize

[in, out] short BlockSize

Размер блока в метке в байтах.

5.1.6.2.7 BlocksCount

[out] short BlocksCount

Общее количество блоков в метке = **SectorsCount** * **BlocksPerSector**.

5.1.6.2.8 LastReader

[in, out] XID_Device LastReader

Последнее устройство, производившее операции над меткой.

При установке данного свойства счетчик ссылок на **LastReader** увеличивается на 1 до удаления объекта метки, либо до назначения нового считывателя.

5.1.6.2.9 GetSector

```
[out] IXID_TagSector GetSector
(
    short SectorIndex
)
```

Возвращает интерфейс типа **IDispatch**, относящийся к сектору с индексом **SectorIndex**.

Свойства и методы **IXID_TagSector** описаны в подразделах данного раздела.

5.1.6.2.9.1 GetBlock

```
[out] IXID_TagBlock GetBlock
(
    short BlockIndex
)
```

Возвращает интерфейс типа **IDispatch**, относящийся к блоку с индексом **BlockIndex** данного сектора.

Свойства и методы **IXID_TagBlock** описаны в подразделах данного раздела.

```
[in, out] string DataAsHexString
```

Данные выбранных сектора и блока метки в виде строки шестнадцатеричных символов.

```
[in, out] string DataAsTextString
```

Данные выбранных сектора и блока метки в виде символьной строки.

5.1.6.2.9.2 DataAsHexString

```
[in, out] string DataAsHexString
```

Данные выбранного сектора метки в виде строки шестнадцатеричных символов.

5.1.6.2.9.3 DataAsTextString

```
[in, out] string DataAsTextString
```

Данные выбранного сектора метки в виде символьной строки.

5.1.6.2.10 BlocksPerSector

[in, out] short BlocksPerSector

Количество блоков в секторе метки.

5.1.6.2.11 SectorsCount

[in, out] short SectorsCount

Количество секторов в метке.

5.1.6.2.12 AllDataAsHexString

[in, out] string AllDataAsHexString

Данные метки в виде строки шестнадцатеричных символов.

5.1.6.2.13 AllDataAsTextString

[in, out] string AllDataAsTextString

Данные метки в виде символьной строки.

5.1.6.2.14 IsEqualTo

```
[out] boolean IsEqualTo
(
    XID_Tag tag,
    boolean CompareData
)
```

Возвращает **true** если свойства данной метки идентичны свойствам **tag**.
При истинном значении **CompareData,** сравниваются также данные меток.

Объект **tag** должен быть предварительно создан.

5.1.6.3 Типы

5.1.6.3.1 XID_eTagType

enum XID_eTagType

Данный перечислимый тип объединяет типы поддерживаемых библиотекой меток RFID:

xidTagType_Unknown	идентифицирует любую неизвестную метку
xidTagType_ICode1	идентифицирует метку типа ICode1
xidTagType_ISO15693	идентифицирует метку типа ISO15693
xidTagType_EPC_Gen1_Class1_HF	идентифицирует метку типа EPC Gen1 Class1 13.56MHz
xidTagType_EPC_Gen2_Class1_UHF	идентифицирует метку типа EPC Gen2 Class1 865-928MHz
xidTagType_ISO14443A	идентифицирует метку типа ISO14443A
xidTagType_ISO14443B	идентифицирует метку типа ISO14443B

5.1.6.3.2 XID_eSlotErrorCodes

enum XID_eSlotErrorCodes

Данный перечислимый тип объединяет коды ошибок для временных интервалов ответов карт:

xidSlotErr_OK	Нет ошибок
xidSlotErr_IsEmpty	Слот пустой (нет карты)
xidSlotErr_WrongCRC	Ошибка при проверке CRC
xidSlotErr_Collision	Коллизия в слоте
xidSlotErr_TimeOut	Превышено время ожидания ответа
xidSlotErr_UndefinedError	Неизвестная ошибка

5.2 Упрощенные объекты

Упрощенные объекты предназначены для быстрого построения систем идентификации, использующих базовые возможности технологии.

5.2.1 XID_SimpleReadersControl

Объект **XID_SimpleReadersControl** предназначен для упрощения работы с библиотекой и обеспечивает ряд базовых функций для некоторых стандартов карт:

- Обнаружение подключенных считывателей
- Чтение серийных номеров меток
  Выполняется для всех типов меток
- Чтение данных с меток
  Выполняется для следующих стандартов меток:
  - Philips ICode1
  - ICodeSLI(ISO15693)
- Запись данных на метки
  Выполняется для следующих стандартов меток:

  - Philips ICode1
- ICodeSLI(ISO15693)
- Подача звукового и светового сигнала на считыватель

5.2.1.1 Свойства

5.2.1.1.1 Language

[in, out] XID_eLanguage Language

Текущий язык для выдачи сообщений библиотеки, а также описания ошибки.

5.2.1.1.2 LastError

[out] XID_eGeneralErrorCodes LastError

Код ошибки при выполнении последней команды хоста (исполняющего компонента библиотеки).

5.2.1.1.3 LastErrorDescription

[out] string LastErrorDescription

Возвращает описание последней ошибки в виде символьной строки.

5.2.1.2 Методы

5.2.1.2.1 LinkReader

```
LinkReader
(
    XID_Device rdr
)
```

Производит подключение считывателя **rdr** к текущему объекту **XID_SimpleReadersControl**.

5.2.1.2.2 LinkReaders

```
LinkReaders
(
    XID_DevicesList rdrs
)
```

Производит подключение списка считывателей **rdrs** к текущему объекту **XID_SimpleReadersControl**.

5.2.1.2.3 UnlinkReader

```
UnlinkReader
(
    XID_Device rdr
)
```

Производит отключение считывателя **rdr** от текущего объекта **XID_SimpleReadersControl**.

5.2.1.2.4 UnlinkAllReaders

```
UnlinkAllReaders()
```

Производит отключение всех считывателей, подключенных к текущему объекту **XID_SimpleReadersControl**.

5.2.1.2.5 Beep

```
Beep
(
    XID_Device rdr
)
```

Подает звуковой сигнал на считыватель **rdr**.

rdr должен быть предварительно подключен командой **LinkReader**.

5.2.1.2.6 Blink

```
Blink
(
    XID_Device rdr
)
```

Подает световой сигнал на считыватель **rdr**.

rdr должен быть предварительно подключен командой **LinkReader**.

5.2.1.2.7 DetectReaders

```
DetectReaders()
(
    XID_DevicesList rdrs
)
```

Производит поиск поддерживаемых считывателей всех поддерживаемых производителей в системе (может длиться довольно долго, предпочтительнее использовать **DetectVendorReaders**).

Очищает список считывателей **rdrs** и записывает в него найденные считыватели.

rdrs должен быть создан вызывающим приложением заранее.

5.2.1.2.8 DetectVendorReaders

```
DetectVendorReaders
(
    [in] string Vendor
)
```

Производит поиск поддерживаемых считывателей от конкретного производителя **Vendor** в системе (такой поиск предпочтительнее, так как он занимает намного меньше времени, нежели использование **DetectReaders**).

5.2.1.3 Типы

5.2.1.3.1 XID_eLanguage

```
enum XID_eLanguage
```

Данный перечислимый тип объединяет поддерживаемые библиотекой языки (на этих языках может выдаваться текстовое сообщение об ошибке):

xidLanguage_English	Английский
xidLanguage_Russian	Русский

5.2.1.3.2 XID_eGeneralErrorCodes

```
enum XID_eGeneralErrorCodes
```

Данный перечислимый тип объединяет общие коды ошибок при работе с устройствами:

xidErr_OK	Нет ошибок
xidErr_NoCard	Нет карты
xidErr_WrongKey	Неверный ключ
xidErr_WrongMode	Неверный режим
xidErr_ExecutionFailed	Ошибка выполнения команды
xidErr_TimeOut	Вышло время ожидания ответа на команду
xidErr_WrongCRC	Неверный CRC
xidErr_LinkFailed	Ошибка при подключении к устройству

xidErr_UpdateFailed	Ошибка обновления прошивки
xidErr_InvalidParameter	Неверный параметр
xidErr_CommandNotSupported	Команда не поддерживается
xidErr_HostIsBusy	Хост (исполняющий компонент библиотеки) занят
xidErr_TooManyReaders	Слишком много считывателей выбрано
xidErr_CommandIsInProgress	Команда находится в процессе выполнения
xidErr_InvalidValue	Неверное значение
xidErr_UndefinedError	Неопределённая ошибка

5.2.1.4 События

5.2.1.4.1 OnTagEvent

```
OnTagEvent
(
    XID_TagsList TagsList,
    boolean IsInserted
)
```

Данное событие возникает при внесении/извлечении одной или нескольких меток в/из поля выбранных считывателей при открытой сессии группы команд **AutoSearch**.

TagsList - список внесенных/извлеченных меток.

IsInserted - флаг, сигнализирующий, внесены ли метки или извлечены из поля считывателя(ей).

5.2.1.4.2 OnSessionPaused

```
OnSessionPaused()
```

Данное событие возникает при фактической установки приостановленного состояния сессии автопоиска группы команд **AutoSearch**.

5.2.1.4.3 OnSessionStopped

```
OnSessionStopped()
```

Данное событие возникает при фактической остановке выполнения сессии автопоиска группы команд **AutoSearch**.

5.2.1.4.4 OnActivateISO14443A

```
OnActivateISO14443A
(
```

```
        string UID
)
```

Данное событие возникает при обнаружении карты стандарта ISO14443A (Mifare) в поле одного из выбранных считывателей при выполнении команды .ControlX.ComplexFuncs.FullAnticollision, либо при открытой сессии группы команд **AutoSearch**.

UID - строка-идентификатор обнаруженной в поле метки (в фотмате HEX).

5.2.1.4.5 OnActivateISO14443B

```
OnActivateISO14443B
(
        string UID
)
```

Данное событие возникает при обнаружении карты стандарта ISO14443B (Mifare) в поле одного из выбранных считывателей при выполнении команды .ControlX.ComplexFuncs.FullAnticollision, либо при открытой сессии группы команд **AutoSearch**.

UID - строка-идентификатор обнаруженной в поле метки (в фотмате HEX).

5.2.1.4.6 OnActivateISO15693

```
OnActivateISO15693
(
        string UID
)
```

Данное событие возникает при обнаружении карты стандарта ISO15693 (ICode) в поле одного из выбранных считывателей при выполнении команды .ControlX.ComplexFuncs.FullAnticollision, либо при открытой сессии группы команд **AutoSearch**.

UID - строка-идентификатор обнаруженной в поле метки (в фотмате HEX).

5.2.1.5 FindCards

```
IXID_SRC_FindCards FindCards
```

Данный интерфейс типа **IDispatch** доступен через **XID_SimpleReadersControl.FindCards**

При помощи команды **FindCards** производится комплексный поиск карт указанных стандартов при помощи указанного списка считывателей.

Найденные карты помещаются в список карт, передаваемый как параметр при запуске выполнения функции.

5.2.1.5.1 Свойства

5.2.1.5.1.1 ExpectedCardsCount

[in, out] long ExpectedCardsCount

Ожидаемое количество меток (устанавливается перед вызовом функции **FindCards**.

5.2.1.5.1.2 UsedReaders

[in, out] XID_DevicesList UsedReaders

Список считывателей, которые должны будут участвовать в работе с метками.

Перед выполнением функции, **UsedReaders** должны быть подключены к текущему объекту **XID_SimpleReadersControl** посредством вызова функции **LinkReaders**.

5.2.1.5.1.3 ExpectedCardsStandards

[out] IXID_SRC_StandardsSelecter ExpectedCardsStandards

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно установить стандарты меток, над которыми следует выполнять операции.

Свойства **XID_SRC_StandardsSelecter** описаны в подразделах данного раздела.

[in, out] boolean UseICode1

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **Philips ICode1**.

[in, out] boolean UseISO15693

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO15693**.

[in, out] boolean UseEPC_G1C1HF

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **EPC, Gen1, Class1, 13.56MHz**.

[in, out] boolean UseISO14443A

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO14443A**.

[in, out] boolean UseISO14443B

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO14443B**.

5.2.1.5.2 Методы

5.2.1.5.2.1 Execute

Execute
(
XID_TagsList tags
)

Запускает выполнение команды, при этом исходные данные и результаты выполнения соответственно берутся и будут отображены в списке меток **tags**.

5.2.1.6 ReadData

IXID_SRC_ReadData ReadData

Данный интерфейс типа **IDispatch** доступен через **XID_SimpleReadersControl.ReadData**

При помощи команды **ReadData** производится чтение данных из карт, переданных в виде списка карт как параметр запуска выполнения функции.

Считанные данные после выполнения команды сохраняются в свойствах карт в списке карт.

5.2.1.6.1 Свойства

5.2.1.6.1.1 UsedReaders

[in, out] XID_DevicesList UsedReaders

Список считывателей, которые должны будут участвовать в работе с метками.

Перед выполнением функции, **UsedReaders** должны быть подключены к текущему объекту **XID_SimpleReadersControl** посредством вызова функции **LinkReaders**.

5.2.1.6.1.2 ExpectedCardsStandards

[out] IXID_SRC_StandardsSelecter ExpectedCardsStandards

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно установить стандарты меток, над которыми следует выполнять операции.

Свойства **XID_SRC_StandardsSelecter** описаны в подразделах данного раздела.

[in, out] boolean UseICode1

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **Philips ICode1**.

[in, out] boolean UseISO15693

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO15693**.

[in, out] boolean UseEPC_G1C1HF

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **EPC, Gen1, Class1, 13.56MHz**.

[in, out] boolean UseISO14443A

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO14443A**.

[in, out] boolean UseISO14443B

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO14443B**.

5.2.1.6.2 Методы

5.2.1.6.2.1 Execute

```
Execute
(
    XID_TagsList tags
```

)

Запускает выполнение команды, при этом исходные данные и результаты выполнения соответственно берутся и будут отображены в списке меток **tags**.

5.2.1.7 WriteData

IXID_SRC_WriteData WriteData

Данный интерфейс типа **IDispatch** доступен через **XID_SimpleReadersControl.WriteData**

При помощи команды **WriteData** производится запись данных в карты, переданные в виде списка карт как параметр запуска выполнения функции.

Данные для записи в виде символьной строки или строки шестнадцатеричных символов могут быть указаны при помощи свойств **AllDataAsTextString** и **AllDataAsHexString** соответственно.

5.2.1.7.1 Свойства

5.2.1.7.1.1 UsedReaders

[in, out] XID_DevicesList UsedReaders

Список считывателей, которые должны будут участвовать в работе с метками.

Перед выполнением функции, **UsedReaders** должны быть подключены к текущему объекту **XID_SimpleReadersControl** посредством вызова функции **LinkReaders**.

5.2.1.7.1.2 ExpectedCardsStandards

[out] IXID_SRC_StandardsSelecter ExpectedCardsStandards

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно установить стандарты меток, над которыми следует выполнять операции.

Свойства **XID_SRC_StandardsSelecter** описаны в подразделах данного раздела.

[in, out] boolean UseICode1

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **Philips ICode1**.

[in, out] boolean UseISO15693

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO15693**.

[in, out] boolean UseEPC_G1C1HF

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **EPC, Gen1, Class1, 13.56MHz**.

[in, out] boolean UseISO14443A

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO14443A**.

[in, out] boolean UseISO14443B

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO14443B**.

5.2.1.7.1.3 AllDataAsTextString

[in, out] string AllDataAsTextString

Данные для записи в виде символьной строки.

Если **AllDataAsTextString** = "", то данные для записи будут браться из списка меток для каждой метки соответственно.

5.2.1.7.1.4 AllDataAsHexString

[in, out] string AllDataAsHexString

Данные для записи в виде строки шестнадцатеричных символов.

Если **AllDataAsHexString** = "", то данные для записи будут браться из списка меток для каждой метки соответственно.

5.2.1.7.2 Методы

5.2.1.7.2.1 Execute

Execute
(

XID_TagsList tags
)

Запускает выполнение команды, при этом исходные данные и результаты выполнения соответственно берутся и будут отображены в списке меток **tags**.

5.2.1.8 AutoSearch

IXID_SRC_AutoSearch AutoSearch

Данный интерфейс типа **IDispatch** доступен через **XID_SimpleReadersControl.AutoSearch.**

При помощи ряда команд **AutoSearch** производится автоматизированный поиск карт в поле во время выполнения сессии в отдельном программном потоке, при этом изменения состава карт в поле сигнализируются событиями объекта **XID_SimpleReadersControl**.

5.2.1.8.1 Свойства

5.2.1.8.1.1 ExpectedCardsCount

[in, out] long ExpectedCardsCount

Ожидаемое количество меток (устанавливается перед вызовом функции **FindCards**.

5.2.1.8.1.2 UsedReaders

[in, out] XID_DevicesList UsedReaders

Список считывателей, которые должны будут участвовать в работе с метками.

Перед выполнением функции, **UsedReaders** должны быть подключены к текущему объекту **XID_SimpleReadersControl** посредством вызова функции **LinkReaders**.

5.2.1.8.1.3 ExpectedCardsStandards

[out] IXID_SRC_StandardsSelecter ExpectedCardsStandards

Возвращает указатель на интерфейс типа **IDispatch**, посредством которого можно установить стандарты меток, над которыми следует выполнять операции.

Свойства **XID_SRC_StandardsSelecter** описаны в подразделах данного раздела.

[in, out] boolean UseICode1

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **Philips ICode1**.

[in, out] boolean UseISO15693

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO15693**.

[in, out] boolean UseEPC_G1C1HF

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **EPC, Gen1, Class1, 13.56MHz**.

[in, out] boolean UseISO14443A

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO14443A**.

[in, out] boolean UseISO14443B

Данный флаг устанавливается при необходимости выполнять операции над картами стандарта **ISO14443B**.

5.2.1.8.1.4 SessionState

[out] XID_eSRC_AS_SessionState SessionState

Текущее состояние сессии автопоиска карт.

5.2.1.8.1.5 CurrentTagsList

[out] XID_TagsList CurrentTagsList

Текущий список меток в поле.

5.2.1.8.1.6 AutoSwitchRF

[in, out] boolean AutoSwitchRF

Если **true**, то поле считывателя будет автоматически включаться и выключаться при опросе меток.

По умолчанию **true**.

5.2.1.8.2 Методы

5.2.1.8.2.1 StartSession

StartSession()

Запускает выполнение сессии автопоиска.

5.2.1.8.2.2 PauseSession

PauseSession
(
[in] boolean WaitForFullPause
)

Приостанавливает выполнение сессии автопоиска. При возведенном флаге **WaitForFullPause**, дожидается полной приостановки, иначе - возвращается из процедуры сразу, приостановка же произойдет при окончании текущего цикла чтения меток.

После выполнения приостановки сессии, список текущих меток в поле остается без изменений.

5.2.1.8.2.3 ResetSession

ResetSession
(
[in] boolean WaitForFullPause
)

Останавливает выполнение сессии автопоиска. При возведенном флаге **WaitForFullPause**, дожидается полной остановки, иначе - возвращается из процедуры сразу, остановка же произойдет при окончании текущего цикла чтения меток.

После выполнения остановки сессии, список текущих меток в поле очищается.

5.2.1.8.3 Типы

5.2.1.8.3.1 XID_eSRC_AS_SessionState

enum XID_eSRC_AS_SessionState

Данный перечислимый тип объединяет возможные состояния сессии автопоиск:

xid_SRC_AS_Stopped	Остановлена
xid_SRC_AS_Running	Выполняется
xid_SRC_AS_Paused	Приостановлена

5.3 Компонента 1С

В состав библиотеки также входит компонента 1С "**XID_CardReader**" - драйвер для работы с оборудованием **RFID** в среде **1С:Предприятие 8.0.**

5.3.1 Введение

Драйвер для считывателей RFID меток для системы «1С:Предприятие 8» (далее по тексту «драйвер») должен реализовывать общую концепцию использования внешних компонент для взаимодействия с торговым оборудованием, подключаемым к системе «1С:Предприятие 8». Использование данного подхода позволяет упростить процесс подключения нового торгового оборудования с системе «1С:Предприятие 8».

5.3.2 Общие требования

Драйвер должен:

- разработан в виде внешней компоненты (описание механизма создания внешних компонент приведено в документе «Документация по технологии создания внешних компонент» на диске ИТС, а также может быть выслано по запросу);
- реализовывать описанную ниже функциональность в полном объеме;
- предоставлять определенный набор свойств и методов для возможности настройки и управления драйвером.

Названия свойств и методов должны быть русскоязычными и англоязычными (например, Порт - Port, Скорость - Speed, Сигнал() – Signal()).

Также драйвер должен поддерживать множественные подключения (в соответствии с идеологией COM технологии), а также поддерживать работу с несколькими устройствами.

При возникновении ошибки в работе драйвер должен сообщить об этом с описанием произошедшей ошибки.

5.3.3 Описание работы с драйвером

5.3.3.1 Начало работы с драйвером

Работа с драйвером начинается с того, что сначала система «1С:Предприятие 8» (в дальнейшем «система») инициализирует соединение с драйвером и загружает его. Далее устанавливаются конфигурационные параметры, необходимые для работы, и происходит подключение к устройству.

Система переводит драйвер в активный режим и после этого от драйвера уже могут поступать сообщения о наступлении внешнего события при появлении меток в поле считывателя. При этом передача данных о метках не ведется.

Работа с драйвером предполагается в активном и пассивном режимах. Пассивный режим подразумевает, что драйвер выполняет какие-либо действия в ответ на команды от системы. При активном режиме работы драйвер должен сам послать сообщение о внешнем событии по

© 2011 MicroEM

изменению своего внутреннего состояния.

Дополнительно к активному режиму для чтения данных из меток введено понятие сессии.

Сессия – это процесс чтения и накопления данных из обнаруженных меток и передача их системе. Начало и окончание сессии определяется системой, вызовом функций драйвера.

При активном режиме или открытой сессии выполнение команд пассивного режима, а также попытки открыть еще одну сессию, для того же логического устройства должны блокироваться и драйвер должен возвращать сообщение об ошибке (команды общего назначения выполняются независимо от режима и состояния сессии).

Работа с метками может осуществляться на чтение и на запись данных.

5.3.3.2 Чтение данных из меток

При пассивном режиме:

1. Дается команда на обнаружение меток в поле считывателя. В ответ возвращается количество обнаруженных RFID меток.
2. Последовательно вызывается функция на получение данных из каждой обнаруженной метки. Драйвер возвращает данные из меток. Команда посылается N раз, где N – количество обнаруженных меток.

При открытой сессии:

1. Система дает команду на начало сессии на чтение.
2. При появлении меток, или если метки уже находятся, в поле считывателя драйвер должен считать данные из них и передать наверх системе. Определены следующие ситуации, когда следует передавать полученные данные из меток:

a. Достигнуто максимальное количество меток, установленное в свойствах компоненты;
b. Нет больше меток в поле считывателя;
c. В поле считывателя ненулевое количество меток и их количество не меняется в течение промежутка времени, установленного в свойствах компоненты;
d. Команда от системы на окончание сессии – в штатном режиме все метки к моменту окончания заполнения документа покинут поле считывателя и передача данных из меток не понадобится.

Выше перечисленные внутренние события драйвера, по наступлению которых ему необходимо передать метки. Все 4 подпункта порождают одно и тоже внешнее событие «ЧтениеМеток» (будет описано ниже). В течение сессии драйвер должен передать данные по каждой метке один единственный раз, т.е. драйвер должен «помнить», какие метки он уже передал.

После отправки события драйвер должен приостановить отправку следующего и ожидать команды сверху о возможности продолжения отправки событий. Если результат обработки события положительный, то драйвер инициирует отправку следующего события, в противном случае требуется перепосылка последнего события, того, после которого драйвер приостановился. То есть из старых событий необходимо хранить только последнюю сформированную строку для отправки (при этом в драйвере при поступлении новые данные должны буферизироваться и высылаться по мере освобождения системы). Пример последовательности такой:

- Отправка События 1;
- Получен ответ СобытиеОбработано(Истина);
- Отправка События 2;
- Получен ответ СобытиеОбработано(Истина);
- Отправка События 3;
- Получен ответ СобытиеОбработано(Ложь);
- Отправка События 3;
- Получен ответ СобытиеОбработано(Ложь);
- Отправка События 3;
- Получен ответ СобытиеОбработано(Истина);
- Отправка События 4;

- Получен ответ СобытиеОбработано(Истина);

3. По окончании процесса приема меток система дает команду на окончание сессии чтения. При этом драйвер не должен более передавать данные о метках наверх и не запоминать их. Также драйвер должен очистить информацию об обнаруженных метках, накопленных в течение сессии.

5.3.3.3 Запись данных в метки

Запись меток протекает в пассивном режиме:

1. Посылается команда на отключение активного режима, если был включен.
2. Дается команда на обнаружение меток в поле считывателя. В ответ возвращается количество обнаруженных RFID меток.
3. Вызывается функция на запись данных в метки. Драйвер записывает переданные данные во все обнаруженные метки. При этом производит проверку, что все метки были успешно записаны, в противном случае возвращает ошибку записи, и передает количество успешно прописанных меток.
4. Восстанавливается предыдущий режим.

5.3.3.4 Состояния переходов между режимами

Состояния могут изменяться только последовательно:

«Пассивный» у «Активный» у «Сессия».

Все прочие попытки некорректного изменения состояния должны приводить к ошибке.

Например, нельзя из пассивного режима открыть сессию, переключить в пассивный режим пока открыта сессия.

5.3.3.5 Завершение работы

По окончании работы с драйвером система закрывает соединение с устройством.

Соединение с драйвером закрывается только при завершении работы системы, либо при пересоздании объекта, связанного с драйвером.

5.3.4 Внешние события от драйвера

Во время подключения драйвер возвращает параметр "Идентификатор устройства". Данный параметр используется в дальнейшем драйвером для указания источника события (устройства) при отправке события, а также конфигурациями для определения связанной с событием обработки обслуживания. Идентификатор устройства должен быть уникальным для каждого подключения, для этого предлагается вести сопоставление «Подключенное устройство – Идентификатор устройства», добавляя новые строки при подключении к устройству и удаляя при отключении от него. Т.е. драйвер при вызове метода Подключить() должен сформировать новый идентификатор устройства (предлагается в качестве имени использовать GUID в виду простоты его получения программным путем) и передать в выходном параметре.

При наступлении событий в драйвере в систему можно посылать сообщение о возникновении внешнего события. При посылке сообщения о внешнем событии необходимо указать: источник внешнего события (Идентификатор устройства), наименование (тип) события и данные.

© 2011 MicroEM

5.3.6 Методы

5.3.6.1 ПолучитьОшибку

```
число
ПолучитьОшибку
(
    [out] строка ОписаниеОшибки
)
```

Возвращает код и описание последней произошедшей ошибки.

5.3.6.2 Подключить

```
булево
Подключить
(
    [in] объект МассивЗначений,
    [out] строка ИДУстройства
)
```

Подключает устройство для использования.

Параметры:
МассивЗначений - массив значений для настройки подключения;
ИДУстройства - идентификатор устройства.

5.3.6.2.1 МассивЗначений

Это объект среды 1С, содержащий параметры подключения учтройства.

5.3.6.2.1.1 МодельУстройства

```
строка МодельУстройства
```

Модель устройства.

5.3.6.2.1.2 Порт

```
число Порт
```

Номер последовательного порта (для считывателя с интерфейсом «RS232» или «Виртуальный RS232»), порта TCP (для считывателя с интерфейсом «TCP/IP») или индекс (начиная от 0) устройства на шине USB (для устройств USB).

5.3.6.2.1.3 АдресНаШине

число АдресНаШине

Адрес устройства на шине (для считывателей, подключаемых по интерфейсу RS485).

0 – задается при единственном устройстве на шине, а также при типе интерфейса связи, отличном от RS485.

В противном случае, для доступа к конкретному устройству на шине, ставится адрес устройства.

5.3.6.2.1.4 АдресIP

строка АдресIP

IP-адрес устройства (для считывателя с интерфейсом «TCP/IP»), иначе “”.

5.3.6.2.1.5 ИнтерфейсСвязи

число ИнтерфейсСвязи

Тип связи для данного подключенного устройства:

- 0: RS232;
- 16: виртуальный RS232;
- 17: виртуальный RS485;
- 32: USB;
- 48: TCP/IP.

5.3.6.2.1.6 НакапливатьМетки

булево НакапливатьМетки

Если установлен в "Ложь", то события в сессии будут возникать по существующему алгоритму. Если установлен в "Истина", то компонентой будет вестись список уникальных номеров меток, который будет пополняться в случае, если при следующем опросе найден номер метки, которого еще нет в этом списке. А уже по этому списку будет смотреться, изменялись ли внесенные в поле метки в течение Timeout или достигло ли их количество значения CardsLimit.

5.3.6.2.1.7 РазделительМетки

строка РазделительМетки

Символ, для разделения идентификатора метки и данных при передаче строки в событии «ЧтениеМеток».

© 2011 MicroEM

5.3.6.2.1.8 ПредельноеКолвоМеток

число ПредельноеКолвоМеток

Предельное количество меток, после которого генерируется событие «ЧтениеМеток».

5.3.6.2.1.9 Таймаут

число Таймаут

Таймаут, после которого при отсутствии изменения количества меток посылается событие (в миллисекундах).

5.3.6.2.1.10 МаксКоличМеток

число МаксКоличМеток

Максимальное количество меток, передаваемое в событии.

5.3.6.3 Отключить

```
булево
Отключить
(
    [in] строка ИДУстройства
)
```

Отключает устройство.

Параметры:
ИДУстройства - идентификатор устройства.

5.3.6.4 ВключитьАктивныйРежим

```
булево
ВключитьАктивныйРежим
(
    [in] строка ИДУстройства
)
```

Включает активный режим.

Параметры:

ИДУстройства - идентификатор устройства.

5.3.6.5 ВыключитьАктивныйРежим

```
булево
ВыключитьАктивныйРежим
(
    [in] строка ИДУстройства
)
```

Выключает активный режим.

Параметры:
ИДУстройства - идентификатор устройства.

5.3.6.6 ОткрытьСессию

```
булево
ОткрытьСессию
(
    [in] строка ИДУстройства
)
```

Начинает сессию получения меток (из всех обнаруженных меток прочитанные данные передаются системе). Если сессия уже открыта, возвращает ошибку.

Параметры:
ИДУстройства - идентификатор устройства.

5.3.6.7 ЗакрытьСессию

```
булево
ЗакрытьСессию
(
    [in] строка ИДУстройства,
    [in] булево ИгнорироватьМеткиВПоле,
    [out] число КоличествоМеток
)
```

Заканчивает сессию работы с метками. Сессия, при наличии меток в поле считывателя, закрывается, только если параметр **ИгнорироватьМеткиВПоле** (IgnoreCardsInField) имеет значение Истина. Если сессия не была закрыта, возвращается ошибка.

Параметры:
ИДУстройства - идентификатор устройства;

© 2011 MicroEM

ИгнорироватьМеткиВПолеесли Ложь, то при наличии в поле меток сессия не закрывается, в противном случае сессия остается открытой;
КоличествоМеток- количество меток в поле при закрытии сессии.

5.3.6.8 СессияОткрыта

```
булево
СессияОткрыта
(
    [in] строка ИДУстройства
)
```

Проверяет, была ли открыта сессия на чтение/запись.

Параметры:
ИДУстройства - идентификатор устройства.

5.3.6.9 НайтиМетки

```
булево
НайтиМетки
(
    [in] строка ИДУстройства,
    [out] число КоличествоМеток
)
```

Осуществляет поиск меток в поле считывателя и возвращает количество обнаруженных меток.

Параметры:
ИДУстройства - идентификатор устройства;
КоличествоМеток- количество обнаруженных меток.

5.3.6.10 СчитатьСледДанные

```
булево
СчитатьСледДанные
(
    [in] строка ИДУстройства,
    [out] строка ИДМетки,
    [out] строка ДанныеМетки
)
```

Считать по-очереди из каждой метки данные.

Параметры:

ИДУстройства - идентификатор устройства;
ИДМетки - идентификатор метки;
ДанныеМетки - данные, записанные в метке.

5.3.6.11 ЗаписатьДанные

```
булево
ЗаписатьДанные
(
    [in] строка ИДУстройства,
    [in] строка Данные,
    [out] число Количество
)
```

Записывает переданные данные в метки, находящиеся в поле считывателя.

Параметры:
ИДУстройства - идентификатор устройства;
Данные - данные для записи в метки;
Количество - кол-во успешно прописанных меток.

5.3.6.12 СобытиеОбработано

```
булево
СобытиеОбработано
(
    [in] строка ИДУстройства,
    [in] булево ФлагСостояния
)
```

Функция устанавливает флаг в «Истина», если событие было успешно обработано, и в «Ложь» в противном случае.

Параметры:
ИДУстройства - идентификатор устройства,
ФлагСостояния - флаг обработки события.

5.3.6.13 ПодатьСигнал

```
булево
ПодатьСигнал
(
    [in] строка ИДУстройства,
```

© 2011 MicroEM

```
        [in] число Длительность
)
```

Подает сигнал на устройстве длительностью, указанной в параметре (если устройство поддерживает данную возможность).

Параметры:
ИДУстройства - идентификатор устройства,
Длительность - длительность сигнала в мс.

5.3.6.14 ПоддерживаетПБ

```
булево
ПоддерживаетПБ
(
        [in] строка ИДУстройства
)
```

Данная функция проверяет, поддерживает ли оборудование работу с противокражным битом.

Параметры:
ИДУстройства - идентификатор устройства.

5.3.6.15 УстановитьПБ

```
булево
УстановитьПБ
(
        [in] строка ИДУстройства,
        [in] строка АдресМетки,
        [in] булево Состояние
)
```

Функция устанавливает состояние противокражного бита в состояние установлен/снят в зависимости от передаваемого значения. Если Адрес метки передается пустой строкой, то установка бита происходит у всех меток в поле считывателя.

Параметры:
ИДУстройства - идентификатор устройства;
АдресМетки - идентификатор метки;
Состояние - Режим установки/снятия противокражного бита.

5.3.6.16 ПолучитьПБ

```
булево
ПолучитьПБ
(
    [in] строка ИДУстройства,
    [in] строка АдресМетки,
    [out] булево Состояние
)
```

Функция получает состояние противокражного бита.

Параметры:
ИДУстройства - идентификатор устройства;
АдресМетки - идентификатор метки;
Состояние - режим получения противокражного бита.

5.3.6.17 ТестУстройства

```
булево
ТестУстройства
(
    [in] объект МассивЗначений,
    [out] строка ДополнительноеОписание
)
```

Выполняет пробное подключение и опрос устройства. При успешном выполнении подключения в описании возвращается описание устройства. При отрицательном результате возвращается описание возникшей проблемы при подключении.

Параметры:
МассивЗначений - массив значений для настройки подключения;
ДополнительноеОписание - Описание результата выполнения теста.

5.3.6.17.1 МассивЗначений

Это объект среды 1С, содержащий параметры подключения учтройства.

5.3.6.17.1.1 МодельУстройства

```
строка МодельУстройства
```

Модель устройства.

5.3.6.17.1.2 Порт

число Порт

Номер последовательного порта (для считывателя с интерфейсом «RS232» или «Виртуальный RS232»), порта TCP (для считывателя с интерфейсом «TCP/IP») или индекс (начиная от 0) устройства на шине USB (для устройств USB).

5.3.6.17.1.3 АдресНаШине

число АдресНаШине

Адрес устройства на шине (для считывателей, подключаемых по интерфейсу RS485).

0 – задается при единственном устройстве на шине, а также при типе интерфейса связи, отличном от RS485.

В противном случае, для доступа к конкретному устройству на шине, ставится адрес устройства.

5.3.6.17.1.4 АдресIP

строка АдресIP

IP-адрес устройства (для считывателя с интерфейсом «TCP/IP»), иначе “”.

5.3.6.17.1.5 ИнтерфейсСвязи

число ИнтерфейсСвязи

Тип связи для данного подключенного устройства:

- 0: RS232;
- 16: виртуальный RS232;
- 17: виртуальный RS485;
- 32: USB;
- 48: TCP/IP.

5.3.6.17.1.6 НакапливатьМетки

булево НакапливатьМетки

Если установлен в "Ложь", то события в сессии будут возникать по существующему алгоритму. Если установлен в "Истина", то компонентой будет вестись список уникальных номеров меток, который будет пополняться в случае, если при следующем опросе найден номер метки, которого еще нет в этом списке. А уже по этому списку будет смотреться, изменялись ли внесенные в поле метки в течение Timeout или достигло ли их количество значения CardsLimit.

5.3.6.17.1.7 РазделительМетки

строка РазделительМетки

Символ, для разделения идентификатора метки и данных при передаче строки в событии «ЧтениеМеток».

5.3.6.17.1.8 ПредельноеКолвоМеток

число МаксКоличМеток

Максимальное количество меток, передаваемое в событии.

5.3.6.17.1.9 Таймаут

число Таймаут

Таймаут, после которого при отсутствии изменения количества меток посылается событие (в миллисекундах).

5.3.6.17.1.10 МаксКоличМеток

число МаксКоличМеток

Максимальное количество меток, передаваемое в событии.

5.3.7 События

5.3.7.1 ОшибкаВДрайвере

Данные – описание ошибки.

Событие присылается, если в процессе работы драйвера возникли ошибки.

5.3.7.2 МеткаВПоле

Данные – описание события.

Событие присылается, только если не открыта сессия, но драйвер находится в активном режиме, и в поле считывателя появились одна или несколько меток. Событие высылается однократно при появлении первой метки в поле считывателя. Повторная посылка события возможна, только после исчезновения всех меток из поля считывателя.

Предполагается, что событие будет использоваться в прикладном решении для выдачи сообщения пользователю, что неоткрыта сессия.

© 2011 MicroEM

5.3.7.3 НетМетокВПоле

Событие присылается только в активном режиме, если из поля исчезли все метки.

5.3.7.4 ЧтениеМеток

Событие посылается только при открытой сессии.

При получении данных от драйвера в виде строки необходимо их разобрать, проверить и погрузить в структуру для удобства работы прикладного уровня. Наиболее подходящей для этих целей будет структура данных «Соответствие», где Ключу сопоставляется уникальный идентификатор метки, а в поле Значение записываются данные, считанные из метки.
Для передачи данных используется тип данных «Строка», поэтому ниже будет определена структура передаваемой строки с целью упрощения дальнейшего разбора.

Для драйвера определена следующая структура строки для передачи данных из меток:
<ИДМетки><РазделительМетки><Данные>#13<ИДМетки><РазделительМетки><Данные>#13<ИДМетки><РазделительМетки><Данные>#13 и т.д.
(<CardID><CardDelimiter><Data>#13<CardID><CardDelimiter><Data>#13<CardID><CardDelimiter><Data>#13 etc.), где ‘<CardDelimiter>’ (настраиваемый параметр) и ‘#13’ (символ перевода строки) – специальные символы-разделители.

Максимальное количество меток в строке – количество, передаваемое в параметре при открытии сессии. Определяет то количество меток, которое прикладное решение готово отработать в одном событии.

Глава

VI

6 Порядок взаимодействия объектов библиотеки при решении различных задач

При построении систем идентификации, использующих возможности данной библиотеки, требуется соблюдать определенный порядок действий в работе с ее объектами.

Ниже изложены варианты взаимодействия объектов библиотеки в зависимости от решаемой задачи.

6.1 Общие принципы

После того как вы установили библиотеку на систему и подключили ее к проекту, можно создавать объекты.

Использование библиотеки зависит от типа решаемых задач.

В случае, если требуется просто считать уникальный номер метки или группы меток, не вникая в тонкости стандартов, а так же выполнить запись/чтение данных, которые можно представить в виде строки символов или шестнадцатеричных цифр, то наилучшим вариантом будет использование т.н. **Упрощенных объектов**.

Если же требуется в полной мере использовать возможности некоторого отдельно взятого стандарта или группы стандартов, то вы подключаете к работе **Базовые объекты**.

6.2 Работа с базовыми объектами

Здесь приводится порядок работы с объектами из группы **Базовые объекты**.

Начать следует с объекта типа **XID_Manager**.
Данный объект будет отвечать за поиск доступных на компьютере считывателей и за хранение их списка.

Рекомендуется использовать один такой объект в приложении (не более - не менее).

После обнаружения устройств при помощи функции **DetectDevices** их список следует загрузить в заранее созданный объект типа **XID_DevicesList** посредством функции **GetDevices**

Допустим основной объект **XID_Manager** уже создан, произведен поиск устройств и их список загружен в **XID_DevicesList**.

Теперь следует создать т.н. базовый объект управления **XID_ControlX**.

Рекомендуется использовать один такой объект на отдельную нить процесса.

Возможна параллельная работа с несколькими устройствами, размещенными на разных портах компьютера в случае если создать два и более объектов управления (по одному на отдельную нить процесса).

Для работы с отдельным устройством с использованием созданного объекта управления

XID_ControlX, требуется выполнить следующие действия:

- Выбрать соответствующее устройство из списка устройств XID_DevicesList при помощи процедуры SelectDevice;
- Подключиться к выбранному устройству через команду LinkDevice.

Теперь можно вызывать функции для работы с конкретным типом устройства или с любым устройством (Any Device Functions), в числе которых входит, например, подача звукового или светового сигнала.

В случае, если требуется произвести поиск меток, запись или чтение данных, то сперва необходимо включить поле считывателя командой RF_ON.

Когда работа с метками завершена, поле может быть выключено командой RF_OFF.

По окончании работы с устройством, последнее отключается процедурой UnlinkDevice.

Если требуется обращение к нескольким устройствам через один объект управления, то нужные устройства можно последовательно выбирать при помощи процедуры SelectDevice, после чего команды объекта управления будут адресованы именно выбранному устройству.

Процедуру SelectDevice можно вызывать в любом месте алгоритма.

6.3 Работа с упрощенными объектами

Здесь приводится порядок работы с объектами из группы Упрощенные объекты.

Первым делом следует создать т.н. упрощенный объект управления XID_SimpleReadersControl.

Рекомендуется использовать один такой объект на отдельную нить процесса.

Вызвать функцию обнаружения считывателей DetectReaders - их список будет загружен в заранее созданный объект типа XID_DevicesList.

Возможна параллельная работа с несколькими устройствами, размещенными на разных портах компьютера в случае если создать два и более объектов управления (по одному на отдельную нить процесса).

Для работы с отдельным устройством с использованием созданного объекта управления XID_SimpleReadersControl, требуется выполнить следующие действия:

- Подключиться к нужному устройству через команду LinkReader или к списку используемых устройств через команду LinkReaders.

Теперь можно вызывать функции сигнализации, чтения карт, а также чтения и записи данных на карты.

При вызове вышеупомянутых функций требуется задать в качестве входного параметра считыватель или список считывателей, которые будут участвовать в процессе выполнения команды. Данные устройства должны быть предварительно подключены командой **Link**... (см. выше)

Перед вызовом таких функций как FindCards, ReadData, WriteData, должен быть создан

объект **XID_TagsList**, который затем будет передан функциям в качестве параметра.

По окончании работы со считывателем или списком считывателей, последнее отключается соответственно либо процедурой **UnlinkReader**, либо **UnlinkAllReaders**.

Глава

VII

7 Примеры использования библиотеки

Здесь собраны примеры кодов на языке **Visual Basic**, где демонстрируется работа с объектами библиотеки

7.1 Примеры работы с базовыми объектами

7.1.1 Пример работы с метками стандарта ISO15693

```
    Dim mgr As New XID_DesignerLib.XID_Manager()
    Dim devs As New XID_DesignerLib.XID_DevicesList
    Dim cx As New XID_DesignerLib.XID_ControlX

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        mgr.DetectDevices()

        mgr.GetDevices(devs)

        Dim ss As New String("")

        If devs.DevicesCount > 0 Then
            MsgBox(devs.DevicesCount, MsgBoxStyle.OkOnly, "Found readers")
            ss = devs.Devices(0).Model
        End If

    End Sub

   Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click


        If devs.DevicesCount = 0 Then Exit Sub

        cx.SelectDevice(devs.Devices(0))

        cx.LinkDevice()

        cx.QueryDevices.Reader.RF_ON.do_RF_ON()



        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.Anticollision.in_AddAFI = False
        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.Anticollision.in_AFI = 0
        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.Anticollision.in_MaskLengthInBits = 0
        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.Anticollision.in_Use1TimeSlot = True
```

```
        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.Anticollision.do_Anticollision()

        MsgBox(cx.LastErrorDescription)

        Dim s = cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.Anticollision.out_UIDs(0)

        Dim a = cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.Anticollision.out_arrUIDs
        Dim t = cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.Anticollision.out_arrTimeSlotsStatuses

        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.ReadBlock.in_UID = _
        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.Anticollision.out_UIDs(0)

        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.ReadBlock.in_SelectionType = XID_DesignerLib.
XID_eIS015693_QueryTypes.xidISO15693_QueryByUID
        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.ReadBlock.in_BlockAddress = 5

        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.ReadBlock.do_ReadBlock()


        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.WriteBlock.in_BlockAsHexString = "AABBCCDD"

        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.WriteBlock.do_WriteBlock()


        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.ReadBlock.do_ReadBlock()

        Dim bl = cx.QueryDevices.Reader.TagStandards.Philips.ICode.
ICodeSLI.ISO15693.ReadBlock.out_BlockAsHexString


        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.ReadBlock.in_BlockAddress = 0
        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.ReadBlocks.in_BlocksCount = 5

        cx.QueryDevices.Reader.TagStandards.Philips.ICode.ICodeSLI.
ISO15693.ReadBlocks.do_ReadBlocks()

        Dim bls = cx.QueryDevices.Reader.TagStandards.Philips.ICode.
ICodeSLI.ISO15693.ReadBlocks.out_arrBlocksAsHexStrings
```

```
        cx.QueryDevices.Reader.RF_OFF.do_RF_OFF()

        cx.UnlinkDevice()

    End Sub
```

7.2 Примеры работы с упрощенными объектами

7.2.1 Пример работы с объектом XID_SimpleReadersControl

```
Public Class Form1
    Dim src As New XID_DesignerLib.XID_SimpleReadersControl
    Dim devs As New XID_DesignerLib.XID_DevicesList
    Dim tags As New XID_DesignerLib.XID_TagsList

    Private Sub DetectDevices_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles DetectDevices.Click
        Dim i
        Me.Cursor = Cursors.WaitCursor

        src.DetectReaders(devs)

        DevicesList.Items.Clear()

        For i = 0 To devs.DevicesCount - 1
            DevicesList.Items.Add(devs.Devices(i).Model)
            DevicesList.Items(i).SubItems.Add(devs.Devices(i).ID)
            DevicesList.Items(i).SubItems.Add(devs.Devices(i).ComPort)
        Next i

        Me.Cursor = Cursors.Default

    End Sub

    Private Sub FindTags_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles FindTags.Click
        Dim i
        Me.Cursor = Cursors.WaitCursor

        src.LinkReaders(devs)

        src.FindCards.ExpectedCardsCount = 10
        src.FindCards.ExpectedCardsStandards.UseISO14443A = True
        src.FindCards.ExpectedCardsStandards.UseISO14443B = True
        src.FindCards.ExpectedCardsStandards.UseISO15693 = True

        src.FindCards.UsedReaders = devs

        src.FindCards.Execute(tags)

        TagsList.Items.Clear()
```

```
        For i = 0 To tags.TagsCount - 1
            TagsList.Items.Add(tags.Tags(i).TypeStr)
            TagsList.Items(i).SubItems.Add(tags.Tags(i).ID)
            TagsList.Items(i).SubItems.Add("")
        Next

        src.UnlinkAllReaders()

        Me.Cursor = Cursors.Default
    End Sub

    Private Sub Beep_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Beep.Click
        Dim i
        Me.Cursor = Cursors.WaitCursor

        src.LinkReaders(devs)


        For i = 0 To devs.DevicesCount - 1
            src.Beep(devs.Devices(i))
        Next i


        src.UnlinkAllReaders()

        Me.Cursor = Cursors.Default
    End Sub

    Private Sub ReadTags_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ReadTags.Click
        Dim i
        Me.Cursor = Cursors.WaitCursor

        src.LinkReaders(devs)

        src.ReadData.ExpectedCardsStandards.UseISO14443A = True
        src.ReadData.ExpectedCardsStandards.UseISO14443B = True
        src.ReadData.ExpectedCardsStandards.UseISO15693 = True

        src.ReadData.UsedReaders = devs

        src.ReadData.Execute(tags)

        For i = 0 To tags.TagsCount - 1
            TagsList.Items(i).SubItems(2).Text = tags.Tags(i).
AllDataAsTextString
        Next

        src.UnlinkAllReaders()

        Me.Cursor = Cursors.Default
```

```
    End Sub

    Private Sub WriteData_Click( ByVal sender As System. Object, ByVal e As
System. EventArgs) Handles WriteData. Click
        Me. Cursor = Cursors. WaitCursor

        src. LinkReaders( devs)

        src. WriteData. ExpectedCardsStandards. UseISO14443A = True
        src. WriteData. ExpectedCardsStandards. UseISO14443B = True
        src. WriteData. ExpectedCardsStandards. UseISO15693 = True

        src. WriteData. UsedReaders = devs

        src. WriteData. AllDataAsTextString = TextData. Text

        src. WriteData. Execute( tags)

        src. UnlinkAllReaders( )

        Me. Cursor = Cursors. Default
    End Sub
End Class
```

Предметный указатель

- 1 -

1C 207

- A -

AccessConditionType 128, 130
Add 54, 186
AddDevicesList 54
AddTagsList 186
Attrib 100
Authenticate 119
AutoSwitchRF 205

- B -

Beep 195
Blink 195
BlocksCount 190
BlockSize 190
BlocksPerSector 192
BSA 57

- C -

Clear 54, 185
CodeKey 121
CommBaudRate 58
CommunicationType 58
ComPort 57
CopyTo 55, 186
CopyValue 127, 129
Count 56, 188
CurrentTagsList 205

- D -

DataAsValue 138
DecrementValue 127, 129
DeleteTagsWithID 187
Destroy 80
DetectDevices 53
DetectReaders 195
DetectVendorDevices 53
DetectVendorReaders 196
DeviceClass 58
do_Attrib 103
do_Authenticate 120
do_CodeKey 122
do_CopyValue 128
do_DecrementValue 127
do_Destroy 81
do_EASAlarm 118
do_GenerateKey 121
do_GetBlocksSecurities 84
do_GetKeysState 76
do_GetNoiseFactor 77
do_GetSystemInfo 86
do_IncrementValue 127
do_LoadKey 121
do_LockEAS 117
do_ReadAllPages 132
do_ReadBusAddress 72
do_ReadConfig 71
do_ReadLock 134
do_ReadOTP 135
do_ReadPages 132
do_ReadTrailer 124
do_ReadValue 125
do_ResetEAS 116
do_ResetRF 75
do_RF_OFF 76
do_RF_ON 76
do_SetEAS 116
do_SetSignalState 77
do_Signal 74
do_UpdateFirmware 74
do_WriteAllPages 133
do_WriteBlock 123
do_WriteBusAddress 73
do_WriteConfig 72
do_WriteDataToCards 70
do_WriteLock 134
do_WriteOTP 135
do_WritePage 133
do_WriteTrailer 125
do_WriteValue 126

- E -

EASAlarm 117

© 2011 MicroEM

ExpectedCardsCount 199, 204
ExpectedCardsStandards 199, 201, 202, 204

- F -

FullAnticollision 64

- G -

GenerateKey 121
GetBlock 191
GetBlocksSecurities 83
GetDeviceIndex 56
GetDevices 53
GetDevicesByClass 55
GetKeysState 76
GetLockBlocked 138
GetNoiseFactor 77
GetReaders 55
GetSector 191
GetSystemInfo 84
GetTagIndex 189
GetTagsByType 187

- H -

HasICreferenceInfo 85
HasMemSizeInfo 85

- I -

IC_ReferenceInfo 86
ICodeEPC_G1C1HF 118
ICodeSLI 115
in_AddendValue 126
in_Address 126
in_AutoSwitchRF 65, 67, 70
in_BusAddress 73
in_Byte9 124
in_CodedTimeSlots 65, 67, 69
in_ConfigLength 71
in_DestinationBlockNumber 127, 128
in_DestroyCode 81
in_FirmwareFileName 74
in_KeyA 124
in_KeyB 124
in_KeyNumber 119, 120
in_KeyType 119, 120
in_NewSignalState 77
in_ResetTime 75
in_SectorNumber 119, 120
in_SourceBlockAddress 126, 127, 128
in_Standards 64, 66, 68
in_StartPageNumber 132
in_SubtrahendValue 127
in_Value 126
IncrementValue 126, 129
inout_arrHigerLevelINF 102
Item 56, 188
IXID_Cmd_Complex_FullAnticollision 64
IXID_Cmd_Complex_ReadDataFromCards 66
IXID_Cmd_Complex_WriteDataToCards 67
IXID_Cmd_Device_ReadBusAddress 72
IXID_Cmd_Device_ReadConfig 71
IXID_Cmd_Device_Signal 73
IXID_Cmd_Device_UpdateFirmware 74
IXID_Cmd_Device_WriteBusAddress 72
IXID_Cmd_Device_WriteConfig 71
IXID_Cmd_Reader_EPC_Anticollision 79
IXID_Cmd_Reader_EPC_Destroy 80
IXID_Cmd_Reader_EPC_WriteBlock 80
IXID_Cmd_Reader_GetKeysState 76
IXID_Cmd_Reader_GetNoiseFactor 77
IXID_Cmd_Reader_ICode_Anticollision 108
IXID_Cmd_Reader_ICode_EAS 109
IXID_Cmd_Reader_ICode_Halt 114
IXID_Cmd_Reader_ICode_ReadSelected 109
IXID_Cmd_Reader_ICode_ReadUnselected 111
IXID_Cmd_Reader_ICode_ResetQuietBit 112
IXID_Cmd_Reader_ICode_WriteBlock 112
IXID_Cmd_Reader_ISO14443A_ActivateIdle 98
IXID_Cmd_Reader_ISO14443A_Halt 98
IXID_Cmd_Reader_ISO14443A_WakeUp 99
IXID_Cmd_Reader_ISO14443B_ActivateIdle 99
IXID_Cmd_Reader_ISO14443B_Attrib 100
IXID_Cmd_Reader_ISO14443B_Halt 103
IXID_Cmd_Reader_ISO14443B_WakeUp 104
IXID_Cmd_Reader_ISO15693_Anticollision 81
IXID_Cmd_Reader_ISO15693_GetBlocksSecurities 83
IXID_Cmd_Reader_ISO15693_GetSystemInfo 84
IXID_Cmd_Reader_ISO15693_LockAFI 86
IXID_Cmd_Reader_ISO15693_LockBlock 86
IXID_Cmd_Reader_ISO15693_LockDSFID 87
IXID_Cmd_Reader_ISO15693_ReadBlock 88

IXID_Cmd_Reader_ISO15693_ReadBlocks 89
IXID_Cmd_Reader_ISO15693_ResetToReady 91
IXID_Cmd_Reader_ISO15693_Select 92
IXID_Cmd_Reader_ISO15693_StayQuiet 92
IXID_Cmd_Reader_ISO15693_WriteAFI 93
IXID_Cmd_Reader_ISO15693_WriteBlock 93
IXID_Cmd_Reader_ISO15693_WriteBlocks 95
IXID_Cmd_Reader_ISO15693_WriteDSFID 96
IXID_Cmd_Reader_MifareStandard_Authenticate 119
IXID_Cmd_Reader_MifareStandard_CodeKey 121
IXID_Cmd_Reader_MifareStandard_CopyValue 127
IXID_Cmd_Reader_MifareStandard_DecrementValue 127
IXID_Cmd_Reader_MifareStandard_GenerateKey 121
IXID_Cmd_Reader_MifareStandard_IncrementValue 126
IXID_Cmd_Reader_MifareStandard_LoadKey 120
IXID_Cmd_Reader_MifareStandard_ReadBlock 122
IXID_Cmd_Reader_MifareStandard_ReadTrailer 123
IXID_Cmd_Reader_MifareStandard_ReadValue 125
IXID_Cmd_Reader_MifareStandard_WriteBlock 123
IXID_Cmd_Reader_MifareStandard_WriteTrailer 124
IXID_Cmd_Reader_MifareStandard_WriteValue 125
IXID_Cmd_Reader_MifareUL_ReadAllPages 131
IXID_Cmd_Reader_MifareUL_ReadLock 133
IXID_Cmd_Reader_MifareUL_ReadOTP 134
IXID_Cmd_Reader_MifareUL_ReadPages 132
IXID_Cmd_Reader_MifareUL_WriteAllPages 133
IXID_Cmd_Reader_MifareUL_WriteLock 134
IXID_Cmd_Reader_MifareUL_WriteOTP 135
IXID_Cmd_Reader_MifareUL_WritePage 132
IXID_Cmd_Reader_ResetRF 75
IXID_Cmd_Reader_RF_OFF 76
IXID_Cmd_Reader_RF_ON 75
IXID_Cmd_Reader_SetSignalState 77
IXID_ComplexFuncs 63
IXID_DeviceFuncs 71
IXID_DeviceTypes 70
IXID_ExtPhilips15693_EASAlarm 117
IXID_ExtPhilips15693_LockEAS 116
IXID_ExtPhilips15693_ResetEAS 116
IXID_ExtPhilips15693_SetEAS 115
IXID_ISO14443B_Card 106
IXID_ISO14443B_Param1 101
IXID_ISO14443B_Param2 101
IXID_ISO14443B_Param3 102
IXID_ISO14443B_Param4 102
IXID_ISO15693_Security 97
IXID_ISO15693_SystemInfo 84
IXID_ISO15693_SystemMemoryInfo 85
IXID_MifareDESfire_Applications 148
IXID_MifareStandard_AccessConditions 128
IXID_MifareStandard_BlockAccessConditions 128
IXID_MifareStandard_SectorTrailerAccessConditions 130
IXID_MifareUL_Pages 135
IXID_SRC_AutoSearch 204
IXID_SRC_StandardsSelecter 199, 201, 202, 204
IXID_Std_EPC_Gen1 78
IXID_Std_EPC_Gen1_Class1 78
IXID_Std_EPC_Global 78
IXID_Std_EPCFuncs 78, 118
IXID_Std_ICodeFuncs 108
IXID_Std_ICodeSLIFuncs 115
IXID_Std_ISO 81
IXID_Std_ISO14443AFuncs 119, 131, 138
IXID_Std_ISO14443BFuncs 99
IXID_Std_ISO15693Funcs 81, 115
IXID_Std_MifareDESfire_Funcs 138
IXID_Std_MifareFuncs 118
IXID_Std_MifareULFuncs 131
IXID_Std_Philips 107
IXID_Std_Philips_ICode 107
IXID_Std_Philips_Mifare 118
IXID_TagBlock 191
IXID_TagSector 191
IXID_TagStandards 78

- L -

LastOperationStatus 190
LastReader 190
LinkReader 194
LinkReaders 194
LoadKey 120
LockBlocked 136
LockEAS 116
LockOfOTP_Blocked 136
LockOfPages10to15_Blocked 136
LockOfPages4to9_Blocked 136

- M -

MemoryInfo 85

© 2011 MicroEM

Mifare 118
MinimumTR0 101
MinimumTR1 101
Model 57
MutualExcludeDuplicates 187

- O -

OnActivateISO14443A 63, 197
OnSessionPaused 197
OnSessionStopped 197
OnTagEvent 197
OTP_Locked 136
out_Address 125
out_BusAddress 72
out_Byte9 124
out_EASSequenceAsHexString 118
out_EPCs 79
out_Is_ISO14443_4_Compatible 98, 99
out_KeyA 123
out_KeyB 124
out_KeysState 76
out_MaxBufLenIndex 103
out_NoiseFactor 77

- P -

PageLocked 136
PagesHexStringArray 137
PauseSession 206
ProtocolType 102

- R -

RawInfoAsHexText 85
ReadAccessBits 130
ReadBusAddress 72
ReadConfig 71
ReadDataFromCards 66
Readers 64, 66, 68
ReadKeyA 130
ReadKeyB 130
ReadTrailer 123
ReadValue 125
Remove 54, 187
RequiredEOF 101
RequiredSOF 101
ResetEAS 116
ResetRF 75
ResetSession 206
RF_OFF 76
RF_ON 75

- S -

SectorsCount 192
SessionState 205
SetEAS 115
SetHexStringToAllTags 188
SetSignalState 77
SetTextStringToAllTags 188
Signal 73
SortByType 188
Standard1k 118
StartSession 206
SubstractTagsList 186
SupportAFI 84
SupportDSFID 84

- T -

Tags 64, 66, 68
Type 189
TypeStr 190

- U -

UnlinkAllReaders 195
UnlinkReader 195
UpdateFirmware 74
UsedReaders 199, 200, 202, 204
UseEPC_G1C1HF 64, 67, 69, 199, 201, 203, 205
UseICode1 65, 67, 69, 199, 201, 202, 204
UseISO14443A 65, 66, 68, 199, 201, 203, 205
UseISO14443B 65, 66, 68, 200, 201, 203, 205
UseISO15693 65, 67, 68, 199, 201, 202, 205

- V -

Version 57

- W -

WriteAccessBits 130
WriteBusAddress 72
WriteConfig 71
WriteDataToCards 67
WriteKeyA 130
WriteKeyB 130
WriteTrailer 124
WriteValue 125

- X -

XID_ControlX 59
XID_Device 56
CloneFrom 56
ID 57
IsEqualTo 58
Vendor 57
XID_eCommBaudRate 59
XID_eCommunicationType 59
XID_eDeviceClass 58
XID_DevicesList 54
XID_eCodedTimeSlots 62, 70
XID_eGeneralErrorCodes 61, 196
XID_eIS015693_QueryTypes 97
XID_eISO14443B_BitRate 106
XID_eISO14443B_MaxFrameSize 105
XID_eISO14443B_MinimumTR0 105
XID_eISO14443B_MinimumTR1 105
XID_eISO14443B_ProtocolType 106
XID_eISO14443B_SlotsCount 104
XID_eLanguage 62, 196
XID_eLedColor 75
XID_eMifareStandardAccessConditionType 129
XID_eMifareStandardAccessType 129
XID_eMifareStandardKeyType 131
XID_eSignalTypes 74
XID_eSlotErrorCodes 61, 193
XID_eSRC_AS_SessionState 206
XID_Manager 53
XID_SimpleReadersControl 193
XID_Tag 189
CloneFrom 189
ID 189
IsEqualTo 192
Vendor 190
XID_eSlotErrorCodes 61, 193
XID_eTagType 192
XID_TagsList 185
XID_UseStandards 64, 66, 68

© 2011 MicroEM