

Marking Builder Plus ActiveX Control for MD-X2000/2500 Series

User's Manual

Read this manual before using the product in order to achieve maximum performance.
Keep this manual in a safe place after reading it so that it can be used at any time.



Introduction

This User's manual provides information regarding usage and precautions in order to control MD-X2000/2500 with Marking Builder Plus ActiveX Control. Read this manual before using the product in order to achieve maximum performance. Keep this manual in a safe place after reading it so that it can be used at any time.

Symbols

In this manual, the following symbols alert you to important messages. Be sure to read the messages.

	Indicates a hazardous situation which, if not avoided, will result in death or serious injury.
	Indicates a hazardous situation which, if not avoided, could result in death or serious injury.
	Indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.
	Indicates a situation which, if not avoided, could result in product damage as well as property damage.
	Indicates cautions and limitations that must be followed during operation.
	Indicates additional information on proper operation.
	Indicates tips for better understanding or useful information.
	Indicates the reference pages and items in this manual.

Cautions

- (1) Unauthorized reproduction of this manual in whole or part is prohibited.
- (2) The contents of this manual may be changed for improvements without prior notice.
- (3) An utmost effort has been made to ensure the contents of this manual are as complete as possible. If there are any mistakes or questions, please contact a KEYENCE office listed in the back of the manual.
- (4) Regardless of item (3), KEYENCE will not be liable for any effect resulting from the use of this unit.
- (5) Any manuals with missing pages or other paging faults will be replaced.

Software License Agreement

NOTICE TO USER: PLEASE READ THIS SOFTWARE LICENSE AGREEMENT (THIS "AGREEMENT") CAREFULLY. BY USING ALL OR ANY PORTION OF THIS SOFTWARE, YOU ARE AGREEING TO BE BOUND BY ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO ANY TERMS OF THIS AGREEMENT, DO NOT USE THIS SOFTWARE.

1. Definition

- 1.1 "use" or "using" means to access, install, download, copy or otherwise benefit from using the functionality of this Software.
- 1.2 "This Software" means the software and all associated documentation provided by KEYENCE.

2. Grant of License.

Conditioned upon compliance with all of the terms and conditions of this Agreement, KEYENCE grants you a nonexclusive and nontransferable license to install this Software on all computers used by your entity in order to use the KEYENCE product. You may make one copy of this Software for backup or archive purposes only.

3. Restrictions.

- 3.1 Except for installation of updates or new functions provided by KEYENCE, you may not modify or add any function to this Software.
- 3.2 You may not reverse engineer, decompile or disassemble this Software.
- 3.3 You may not create derivative works based on this Software.
- 3.4 Other than expressly stated by KEYENCE, you may not resell, retransfer, rent or otherwise redistribute this Software to any third parties.

4. Intellectual Property Rights.

Except as expressly stated herein, KEYENCE reserves all right, title and interest in this Software, and all associated copyrights, trademarks, and other intellectual property rights therein.

5. Disclaimer.

Keyence is licensing this Software to you "AS IS" and without any warranty of any kind. In no event will KEYENCE or its suppliers be liable to you for any damages, claims, costs or any lost profits caused by using this Software.

6. Termination.

- 6.1 Your license under this Agreement will terminate automatically if you destroy this Software and the copy of this Software in your possession or voluntarily return this Software to us.
- 6.2 Your license under this Agreement will terminate automatically without any notice from KEYENCE if you fail to comply with any of the terms and conditions of this Agreement. Promptly upon termination, you shall cease all use of this Software and destroy all copies, full or partial, of this Software in your possession or control.
- 6.3 You will compensate KEYENCE for costs or any lost profits caused by your violation or breach of any term of this Agreement.

7. Governing Law.

- 7.1 This Agreement will be governed by and construed in accordance with the substantive laws of Japan without regards to the principles of conflicts of law.
- 7.2 If any part of this Agreement is found void and unenforceable, it will not affect the validity of the balance of this Agreement, which shall remain valid and enforceable according to its terms and conditions.

Contents

Software License Agreement

Interface List

Class MBActX	6
Class MBComm	7
Class MBJob	7
Class MBBlock	8
Class MBMarkingCondition	10
Class MBEquipment	10
Class MBMatrix	11
Class MBMatrixCell	11
Class MBOperation	11
Class MBError	12
Class MBErrorHistory	12
Class MBControllerFile	13
Class MBDxf	13
Class MBOverprintingCode	13
Class MB3DShape	13
Class MBCommonBlock	13
Class MBWindowCheck	14
Class MBXYTracking	14
Class MBZTracking	14

Before Use

Required devices and operating environment	15
Example of Each Development Language	15
About types	15
How to refer to "MBPlusActiveX"	16
Setting of the Target OS	17
Access to Interfaces in Development Environments	18
File operation from interface	19
Standard Programs	21
Other Notes	22

MBActX

How to Read Interface Table	23
Method	24
ZoomAdjust	24
ZoomAll	24
Zoom	24
ZoomAdjustMatrixCell	24
ShowBlockHighlighting	24
HideBlockHighlighting	24
Redraw	25
OpenJob	25
SaveJob	25
OpenControllerJob	25
SaveControllerJob	26
DeleteControllerJob	26
ReceiveControllerJob	26
SendControllerJob	27
DeleteControllerFile	27
ReceiveControllerFile	27
SendControllerFile	27
ReceiveControllerBufferFile	28
SendControllerBufferFile	28
GetFontSetting	29
SetFontSetting	29
InitMBActX	29
InitMBActXEx	30
Block	30
Matrix	30
Shape	30
ControllerFile	31
MarkingCondition	31
Error	31
ErrorHistory	31
UpdateErrorCount	31

UpdateErrorHistoryCount	31
UpdateControllerFileCount	31
GetWindowCheck	32
GetXYTracking	32
GetZTracking	32
Property	33
Context	33
PreviewOperation	33
Job	33
Operation	33
Equipment	34
Comm	34
CommonBlock	34
LibraryVersion	34
MarkingConditionCount	34
Dxf	34
MarkingUnit	34
IsBlockingCommunication	35
IsAutoRedraw	35
Event	36
EvExpansionStart	36
EvExpansionEnd	36
EvMarkingStart2	36
EvMarkingEnd2	36
EvGuideLaserMarkingStart	37
EvGuideLaserMarkingEnd	37
EvConnectionError	37
EvError	38
Ev2DCodeReadingStart	38
Ev2DCodeReadingEnd	38
EvMarkingConfirmationStart	39
EvMarkingConfirmationEnd	39
EvMarkingWorkflowEnd	39

MBComm

Method	40
Online	40
Offline	40
SendAndReceive	40
Property	41
ConnectionType	41
IpAddress	41
IsOnline	41
UsbControllerSerialNo	42
LibraryVersion	42

MBJob

Method	43
SetTolerance	43
GetTolerance	43
Property	44
Title	44
MoveCondition	44
StationaryTriggerDelay	44
IsMarkingUpperEnergyCheckEnable	44
MarkingUpperEnergyLimit	45
IsMarkingLowerEnergyCheckEnable	45
MarkingLowerEnergyLimit	45
IsContinuousMarkingEnable	45
ContinuousMarkingRepetition	46
ContinuousMarkingInterval	46
WorkpiecePositionReferenceX	46
WorkpiecePositionReferenceY	46
WorkpiecePositionOffsetX	47
WorkpiecePositionOffsetY	47
WorkpiecePositionOffsetZ	47
WorkpiecePositionOffsetAngle	47
WorkpiecePositionZCorrection	48
AutoFocusMeasureCount	48

MBBlock

Method	49	OverprintingCodeCount	89
SetMarkingCondition	49	IsOverprintingCode	90
OverprintingCode	49	ReferencePosition	90
SetDefaultLogoSize	50	SkipCross	91
Property	50	ArcRadius	91
IsEditable	50	StartAngle	92
LaserPower	51	CharSpaceAngle	92
ScanSpeed	51	OpenAngle	93
Frequency	52	FontType	93
SpotVariable	52	FontNo	94
Repetition	53	FontName	94
IsMarkingEnable	53	IsFontStyleBold	95
QualityLevel	54	IsFontStyleItalic	95
ProcessingTime	55	CharLineType	96
MarkingDelayTime	55	MultipleLineWidth	96
Width	56	IsAutoMultipleLines	97
Height	56	MultipleLines	97
IsLogoAspectRatioKeep	57	WobbleLineWidth	98
LogoWidth	57	WobbleOverlapRatio	98
LogoHeight	58	CharFullWidth	99
X	58	CharAngle	99
Y	59	CharAlignType	100
Z	59	IsLaserPowerBlockCommon	100
IsCharSizeRatioFixed	60	IsScanSpeedBlockCommon	101
IsCharSizeProportional	60	IsFrequencyBlockCommon	101
CharWidth	61	IsSpotVariableBlockCommon	102
CharHeight	61	IsQualityLevelBlockCommon	102
CharSpace	62	IsRepetitionBlockCommon	103
CharPitch	63	IsZBlockCommon	103
Angle	63	IsSkipCrossBlockCommon	104
IsFillMarkingConditionEnable	64	IsFillLaserPowerBlockCommon	104
FillLaserPower	64	IsFillScanSpeedBlockCommon	105
FillScanSpeed	65	IsFillFrequencyBlockCommon	105
FillFrequency	65	IsFillSpotVariableBlockCommon	106
FillSpotVariable	66	IsFillRepetitionBlockCommon	106
FillRepetition	67	IsFillLineIntervalBlockCommon	107
FillType	67		
FillLineInterval	68		
FillMargin	68		
FillAngle	69		
FillCrossAngle	69		
FillDirection	70		
FillSkipLineCount	71		
FillStartPosition	71		
IsFillOverprinting	72		
FillOverprinting	72		
FillOverprintingCount	73		
FillBoundaryOffset	73		
FillBoundaryOrder	74		
IsCreateFillBoundary	74		
BarcodeHeight	75		
NarrowBarWidth	75		
BarRatio	76		
QuietZone	76		
BlockType	77		
DataType	78		
Text	79		
IsCheckDigit	80		
IsLocatedOn3DShape	80		
ShapeNo	81		
QRVersion	81		
QRErrorCorrectionLevel	82		
CellSize	82		
SymbolSize	83		
CodeForm	84		
Pattern2D	85		
PatternBarcode	86		
PatternGS1DataBar	87		
LinearCodeHeight	87		
ModuleWidth	88		
CompositeCodeModuleHeight	88		
SeparatorHeight	89		

MBMarkingCondition

Property	108
Caption	108
LaserPower	109
ScanSpeed	109
Frequency	109
SpotVariable	109
Repetition	109
FillLineInterval	109

MBEquipment

Method	110
AdjustTime	110
SynchronizePcTime	110
GetControllerTime	110
StartWarmUp	110
StopWarmUp	111
SetOffsetPower	111
GetOffsetPower	111
SetInstallationPositionCorrection	112
GetInstallationPositionCorrection	112
StartPowerMonitorZeroPointCorrection	112
GetLaserPowerCalibrationMode	113
SetLaserPowerCalibrationMode	113
StartLaserPowerAutoCalibration	114
StopLaserPowerAutoCalibration	114
StartLaserPowerMaintenance	114
StopLaserPowerMaintenance	114
GetLaserPowerMaintenanceOutput	115
SetDistancePointerStatus	115
GetDistancePointerStatus	115
SetCumulativeMarkingCount	116
GetCumulativeMarkingCount	116
SetCumulativeMarkingCountString	116
GetCumulativeMarkingCountString	116
ApplyLaserScanningCalibration	117

Contents

GetOperatingTime	117
GetScannerOperatingTime	117
GetShutterOperatingCount	117
GetLaserSafetyModuleOperatingCount	117
GetMarkingUnitTemperature	118
GetControllerTemperature	118
GetLaserPowerCalibrationResult	118
GetLaserAccumulatedTime	118
GetWindowCheckResult	118
GetStartupWindowCheckEnabled	118
SetStartupWindowCheckEnabled	119
GetWindowCheckAlarmThreshold	119
SetWindowCheckAlarmThreshold	119
GetWindowCheckSensitivity	119
SetWindowCheckSensitivity	119
StartWindowCheck	119
Property	120
LaserMarkerVersion	120
ControllerSerialNo	120
HeadSerialNo	120

MBMatrix

Method	121
MatrixCell	121
SetWidth	121
SetHeight	121
Property	122
RowCount	122
ColumnCount	122
X	122
Y	122
CellWidth	122
CellHeight	122
MatrixWidth	123
MatrixHeight	123

MBMatrixCell

Property	124
OffsetX	124
OffsetY	124
OffsetZ	124
OffsetAngle	125
IsMarkingEnable	125
MarkingDelayTime	125
OriginX	125
OriginY	125

MBOperation

Method	126
SetCurrentCounterValue	126
GetCurrentCounterValue	126
SetCurrentCounterString	127
GetCurrentCounterString	127
SetCurrentCounterRepetition	128
GetCurrentCounterRepetition	128
SetCurrentCounterRepetitionString	129
GetCurrentCounterRepetitionString	129
GetJobTitle	130
GetWorkDistanceGap	130
SetCharString2	130
GetCharString2	130
SetWorkpiecePositionAdjustment	131
GetWorkpiecePositionAdjustment	131
SetWorkpiecePositionAdjustmentZ	132
GetWorkpiecePositionAdjustmentZ	132
StartMarking	132
StartGuideLaserMarking	133
StopMarking	133
GetCurrentJobNo	134
SetCurrentJobNo	134
ClearError	134
BeginTransaction	135

CommitTransaction	135
RollbackTransaction	135
StartCameraScanningMode2	136
FinishCameraScanningMode	136
SetCameraPosition	137
OpenShutter	137
CloseShutter	137
SetMatrixCellOffset	138
GetMatrixCellOffset	138
SetMatrixCellMarkingEnable	139
GetMatrixCellMarkingEnable	139
Read2DCode2	140
Read2DCodePosition2	141
Read2DCodeBlock	141
Get2DCodeReadResult	141
GetCameraViewImage	142
StartZTrackingCalibration	142
CheckZPosition	142
StartZTracking	142
GetZTrackingResult	143
StartXYTracking	143
GetXYTrackingResult	144
GetWindowCheckResult	144
StartWindowCheck	145
SetLighting	145
GetLighting	145
Property	146
IsCameraScanningMode	146
IsExpanding	146
IsError	146
IsWarning	146
IsReady	147
IsMarking	147
IsShutterOpen	147
IsLaserActivated	147
IoEncodedCharacter	148
IsCameraFinderView	148

MBError

Property	149
ErrorCode	149
ErrorCaption	149

MBErrorHistory

Property	150
ErrorCode	150
ErrorCaption	150
ErrorDateTime	150

MBControllerFile

Property	151
FileName	151
JobNo	151

MBDxf

Method	152
Convert	152
Property	152
IsDetectPointElement	152

MBOverprintingCode

Property	153
CodeForm	153
LaserPower	153
IsLaserPowerBlockCommon	153
ScanSpeed	154
IsScanSpeedBlockCommon	154
Frequency	154
IsFrequencyBlockCommon	154
SpotVariable	155
IsSpotVariableBlockCommon	155
Repetition	155
IsRepetitionBlockCommon	155
FillLineInterval	156
IsFillLineIntervalBlockCommon	156
FillMargin	156
PatternBarcode	157
Pattern2D	158

MB3DShape

Property	159
X	159
Y	159
Z	159

MBCommonBlock

Property	160
LaserPower	160
ScanSpeed	160
Frequency	160
SpotVariable	160
Repetition	161
QualityLevel	161
Z	161
FillLineInterval	161
SkipCross	161

MBWindowCheck

Property	162
Enabled	162
AlarmThreshold	162
Sensitivity	162

MBXYTracking

Property	163
Enable	163
CorrectionThreshold	163

MBZTracking

Property	163
Enable	163
Method	163
GetHeightLimit	163
SetHeightLimit	163

Interface List

Class MBackX

Method

ZoomAdjust	Adjusts the preview window to show all blocks.	24
ZoomAll	Displays the entire marking area in the preview window.	24
Zoom	Enlarges the preview window at the specified coordinate and scale ratio.	24
ZoomAdjustMatrixCell	Displays a preview adjusted to the specified cell.	24
ShowBlockHighlighting	Highlights a block in the preview window.	24
HideBlockHighlighting	Cancels the highlight in the preview window.	24
Redraw	Re-draws the preview.	25
OpenJob	Opens a job file in the PC for editing.	25
SaveJob	Saves the job file being edited to the PC.	25
OpenControllerJob	Opens a job saved in the controller for editing.	25
SaveControllerJob	Saves the job being edited to the controller.	26
DeleteControllerJob	Deletes a job from the controller.	26
ReceiveControllerJob	Copies and saves a job in the controller to the PC.	26
SendControllerJob	Copies and saves the job file in the PC to the controller.	27
DeleteControllerFile	Deletes a file from the controller.	27
ReceiveControllerFile	Copies and saves a file in the controller to the PC.	27
SendControllerFile	Copies and saves the target file in the PC to the controller.	27
ReceiveControllerBufferFile	Copies and saves a file in the controller's logo custom character buffer to the PC.	28
SendControllerBufferFile	Directly expands a file in the PC to the controller's logo custom character buffer.	28
GetFontSetting	Gets the font assignment information from the PC.	29
SetFontSetting	Sets the font assignment information in the PC.	29
InitMBackX	Initializes ActiveX and selects a connection destination.	29
InitMBackXEx	Initializes ActiveX in the specified language and selects a connection destination.	30
Block	Gets the block information.	30
Matrix	Gets the matrix object.	30
Shape	Gets the 3D shape setting.	30
ControllerFile	Gets the file object of the controller.	31
MarkingCondition	Gets the marking parameters clipboard object.	31
Error	Gets the list of current errors.	31
ErrorHistory	Gets the error history.	31
UpdateErrorCount	Gets the number of current errors.	31
UpdateErrorHistoryCount	Gets the total number of errors listed in the error history.	31
UpdateControllerFileCount	Gets the number of files in the controller.	31
GetWindowCheck	Gets the window maintenance information object.	32
GetXYTracking	Gets the XY tracking setting information object.	32
GetZTracking	Gets the Z tracking setting information object.	32

Property

Context	Switch between editing mode and operation monitoring mode.	33
PreviewOperation	Changes the cursor operation on the preview.	33
Job	Gets the setting information object.	33
Operation	Gets the operation information object.	33
Equipment	Gets the unit setup object.	34
Comm	Gets the communication management object.	34
CommonBlock	Gets the common parameter object for blocks.	34
MarkingUnit	Gets the model information.	34
LibraryVersion	Gets the DLL version number.	34
MarkingConditionCount	Gets the number of marking conditions clipboards.	34
Dxf	Gets the DXF conversion object.	34
IsBlockingCommunication	Gets/sets whether to communicate with the controller in the blocking communication method.	35
IsAutoRedraw	Gets/sets whether to update the drawing automatically when the property has been changed.	35

● Event

EvExpansionStart	This event occurs when expansion has started.	36
EvExpansionEnd	This event occurs when expansion has been completed.	36
EvMarkingStart2	This event occurs when marking has started.	36
EvMarkingEnd2	This event occurs when marking has been completed.	36
EvGuideLaserMarkingStart	Occurs when guide laser marking has started.	37
EvGuideLaserMarkingEnd	Occurs when guide laser marking has been completed.	37
EvConnectionError	This event occurs when the USB/Ethernet is physically disconnected.	37
EvError	This event occurs when the error occurrence status has changed.	38
Ev2DCodeReadingStart	This event occurs when 2D code reading has started.	38
Ev2DCodeReadingEnd	This event occurs when 2D code reading has been completed.	38
EvMarkingConfirmationStart	Occurs when marking verification function is started.	39
EvMarkingConfirmationEnd	Occurs when marking verification function has been completed.	39
EvMarkingWorkflowEnd	It will occur when the set operation before marking, marking, and operation after marking are completed.	39

Class MBComm

● Method

Online	This method establishes a connection to the controller.	40
Offline	Disconnect the communication.	40
SendAndReceive	Send and receive data online.	40

● Property

ConnectionType	Selects a connection type.	41
IpAddress	Gets/sets the IP address (For Ethernet connection)	41
IsOnline	Gets the connection status.	41
UsbControllerSerialNo	When connected via USB, gets/sets the connected controller serial No.	42
LibraryVersion	Gets the DLL version number.	42

Class MBJob

● Method

SetTolerance	Sets the Z height tolerance for auto-focus and external displacement sensor adjustment.	43
GetTolerance	Gets the Z height tolerance for auto-focus and external displacement sensor adjustment.	43

● Property

Title	Gets a job title.	44
MoveCondition	Gets moving / stationary marking condition.	44
StationaryTriggerDelay	Gets/sets the trigger delay.	44
IsMarkingUpperEnergyCheckEnable	Gets/sets whether to enable the marking energy upper limit threshold.	44
MarkingUpperEnergyLimit	Sets the marking energy upper limit threshold value.	45
IsMarkingLowerEnergyCheckEnable	Gets/sets whether to enable the marking energy lower limit threshold.	45
MarkingLowerEnergyLimit	Sets the marking energy lower limit threshold value.	45
IsContinuousMarkingEnable	Gets/sets whether to perform continuous marking.	45
ContinuousMarkingRepetition	Gets/sets the repetition of continuous marking.	46
ContinuousMarkingInterval	Gets/sets the interval of continuous marking.	46
WorkpiecePositionReferenceX	Gets/sets the movement reference point (X) for workpiece position adjustment.	46
WorkpiecePositionReferenceY	Gets/sets the movement reference point (Y) for workpiece position adjustment.	46
WorkpiecePositionOffsetX	Gets/sets the X coordinate workpiece position adjustment offset amount.	47
WorkpiecePositionOffsetY	Gets/sets the Y coordinate workpiece position adjustment offset amount.	47
WorkpiecePositionOffsetZ	Gets/sets the Z coordinate workpiece position adjustment offset amount.	47
WorkpiecePositionOffsetAngle	Gets/sets the θ angle workpiece position adjustment offset amount.	47
WorkpiecePositionZCorrection	Gets/sets the Z height correction method for workpiece position adjustment.	48
AutoFocusMeasureCount	Gets/sets the number of measurements with auto focus for workpiece position adjustment.	48

Class MBlock

Method

SetMarkingCondition	Sets a block with the marking conditions registered in the marking conditions clipboard.	49
OverprintingCode	Gets the overprinting (multi-pass) settings of a barcode/2D code.	49
SetDefaultLogoSize	Resets the height and width of a logo to the default.	50

Property

IsEditable	MBAcX determines whether a block is editable.	50
LaserPower	Gets/sets the laser power.	51
ScanSpeed	Gets/sets the scan speed.	51
Frequency	Gets/sets the pulse frequency.	52
SpotVariable	Gets/sets the spot variable.	52
Repetition	Gets/sets the repetition.	53
IsMarkingEnable	Gets/sets whether marking is enabled.	53
QualityLevel	Gets/sets the quality level.	54
ProcessingTime	Gets/sets the on time of a fixed point mark.	55
MarkingDelayTime	Gets/sets the wait time for start marking.	55
Width	Gets the width of a block.	56
Height	Gets the height of a block.	56
IsLogoAspectRatioKeep	Gets/sets whether to keep the logo aspect ratio.	57
LogoWidth	Gets/sets the width of a logo.	57
LogoHeight	Gets/sets the height of a logo.	58
X	Gets/sets the X coordinate.	58
Y	Gets/sets the Y coordinate.	59
Z	Gets/sets the Z coordinate.	59
IsCharSizeRatioFixed	Gets whether the character size is specified to fixed ratio.	60
IsCharSizeProportional	Gets whether the character is proportional.	60
CharWidth	Gets/sets the width of a character.	61
CharHeight	Gets/sets the height of a character.	61
CharSpace	Gets/sets the space between characters.	62
CharPitch	Gets/sets character pitch.	63
Angle	Gets/sets the block angle.	63
IsFillMarkingConditionEnable	Gets/sets whether to use the fill marking condition.	64
FillLaserPower	Gets/sets the fill line laser power of a hatch logo.	64
FillScanSpeed	Gets/sets the fill line scan speed of a hatch logo.	65
FillFrequency	Gets/sets the fill line pulse frequency of a hatch logo.	65
FillSpotVariable	Gets/sets the fill line spot variable of a hatch logo.	66
FillRepetition	Gets/sets the fill line repetition of a hatch logo.	67
FillType	Gets/sets the fill type of a hatch logo.	67
FillLineInterval	Gets/sets the fill interval of a hatch logo. The unit is μm .	68
FillMargin	Gets/sets the fill margin of a hatch logo.	68
FillAngle	Gets/sets the fill angle of a hatch logo.	69
FillCrossAngle	Gets/sets the fill tolerance angle of a hatch logo.	69
FillDirection	Gets/sets the fill direction of a hatch logo.	70
FillSkipLineCount	Gets/sets the fill skip line count of a hatch logo.	71
FillStartPosition	Gets/sets the fill start position of a hatch logo.	71
IsFillOverprinting	Gets/sets the fill line overwriting ON/OFF of the hatch logo.	72
FillOverprinting	Gets/sets the fill overprinting direction of a hatch logo.	72
FillOverprintingCount	Gets/sets the fill overprinting count of a hatch logo.	73
FillBoundaryOffset	Gets/sets the fill boundary offset of a hatch logo.	73
FillBoundaryOrder	Gets/sets the fill boundary order type of a hatch logo.	74
IsCreateFillBoundary	Gets/sets whether to create the fill boundary of a hatch logo.	74
BarcodeHeight	Gets/sets the fill overprinting direction of a hatch logo.	75
NarrowBarWidth	Gets/sets the narrow bar width of a barcode.	75
BarRatio	Gets/sets the bar ratio of a barcode.	76
QuietZone	Gets/sets the quiet zone width of a barcode/2D code/GS1Databar.	76
BlockType	Gets the type of a block.	77

DataType	Gets the detailed type of a block.	78
Text	Gets/sets the character string, logo, Barcode or 2D code.	79
IsCheckDigit	Gets/sets the check digit of a barcode.	80
IsLocatedOn3DShape	Gets whether a block is placed on a 3D shape.	80
ShapeNo	Gets the 3D shape setting number.	81
QRVersion	Gets/sets the version of a QR code.	81
QRErrorCorrectionLevel	Gets/sets the error correction level of a QR code.	82
CellSize	Gets/sets the cell size of a 2D code.	82
SymbolSize	Gets/sets the symbol size of a DataMatrix or GS1 DataMatrix.	83
CodeForm	Gets the marking type (Normal/Black/White Inversion/Base) of a 2D/Barcode/GS1DataBar.	84
Pattern2D	Gets/sets the 2D code pattern.	85
PatternBarcode	Gets/sets the barcode pattern.	86
PatternGS1DataBar	Gets/sets the pattern of GS1DataBar.	87
LinearCodeHeight	Gets/sets the height of the linear code of GS1DataBar.	87
ModuleWidth	Gets/sets the module width of GS1DataBar.	88
CompositeCodeModuleHeight	Gets/sets the 2D module height of GS1DataBar.	88
SeparatorHeight	Gets/sets the separator height of GS1DataBar.	89
OverprintingCodeCount	Gets the overprinting count of a barcode/2D code.	89
IsOverprintingCode	Gets whether the overprinting count of a barcode/2D code is set.	90
ReferencePosition	Gets/sets the character string reference position.	90
SkipCross	Gets/sets the skip cross of characters.	91
ArcRadius	Gets/sets the arc radius of a character string arc.	91
StartAngle	Gets/sets the start angle of a character string arc.	92
CharSpaceAngle	Gets/sets the space angle of a character string arc.	92
OpenAngle	Gets/sets the open angle of a character string arc.	93
FontType	Gets/sets the character font type.	93
FontNo	Gets/sets the font number of a character string.	94
FontName	Gets the font name of a character string.	94
IsFontStyleBold	Gets whether the TrueType font is bold.	95
IsFontStyleItalic	Gets whether the TrueType font is italic.	95
CharLineType	Gets/sets the line type of a character string.	96
MultipleLineWidth	Gets/sets the multiple line width of a character string.	96
IsAutoMultipleLines	Gets/sets whether to set the number of multiple lines of a character string to Auto.	97
MultipleLines	Gets/sets the number of multiple marking lines of a character string.	97
WobbleLineWidth	Gets/sets the Wobble line width of a character string.	98
WobbleOverlapRatio	Gets/sets the Wobble overlap rate of a character string.	98
CharFullWidth	Gets/sets the full width of a character string.	99
CharAngle	Gets/sets the character angle.	99
CharAlignType	Gets/sets the character layout (space/angular interval/distribution).	100
IsLaserPowerBlockCommon	Gets/sets whether to use the common laser power parameter for blocks.	100
IsScanSpeedBlockCommon	Gets/sets whether to use the common scan speed parameter for blocks.	101
IsFrequencyBlockCommon	Gets/sets whether to use the common pulse frequency parameter for blocks.	101
IsSpotVariableBlockCommon	Gets/sets whether to use the common spot variable parameter for blocks.	102
IsQualityLevelBlockCommon	Gets/sets whether to use the common quality level parameter for blocks.	102
IsRepetitionBlockCommon	Gets/sets whether to use the common repetition parameter for blocks.	103
IsZBlockCommon	Gets/sets whether to use the common Z coordinate parameter for blocks.	103
IsSkipCrossBlockCommon	Gets/sets whether to use the common skip cross parameter for blocks.	104
IsFillLaserPowerBlockCommon	Gets/sets whether to use the common fill line laser power parameter for blocks.	104
IsFillScanSpeedBlockCommon	Gets/sets whether to use the common fill line scan speed parameter for blocks.	105
IsFillFrequencyBlockCommon	Gets/sets whether to use the common fill line pulse frequency parameter for blocks.	105
IsFillSpotVariableBlockCommon	Gets/sets whether to use the common fill line spot variable parameter for blocks.	106
IsFillRepetitionBlockCommon	Gets/sets whether to use the common fill line repetition parameter for blocks.	106
IsFillLineIntervalBlockCommon	Gets/sets whether to use the common line interval parameter for blocks.	107

Class MBMarkingCondition

Property

Caption	Gets the comments on the marking conditions clipboard.	108
LaserPower	Gets laser power from the marking conditions clipboard.	109
ScanSpeed	Gets scan speed from the marking conditions clipboard.	109
Frequency	Gets frequency from the marking conditions clipboard.	109
SpotVariable	Gets spot variable from the marking conditions clipboard.	109
Repetition	Gets repetition from the marking conditions clipboard.	109
FillLineInterval	Gets the fill interval of the marking conditions clipboard.	109

Class MBEquipment

Method

AdjustTime	Synchronizes with a specified date and time.	110
SynchronizePcTime	Sets the date and time of a laser marker to the date and time of the connected PC.	110
GetControllerTime	Gets the date and time of the controller.	110
StartWarmUp	Forcibly starts warm up.	110
StopWarmUp	Cancels the warm up.	111
SetOffsetPower	Sets the power offset value in unit setup.	111
GetOffsetPower	Gets the power offset value from unit setup.	111
SetInstallationPositionCorrection	Sets the position correction.	112
GetInstallationPositionCorrection	Gets the focal distance and position correction.	112
StartPowerMonitorZeroPointCorrection	Performs the power monitor zero point correction.	112
GetLaserPowerCalibrationMode	Gets laser power calibration mode.	113
SetLaserPowerCalibrationMode	Sets the laser power calibration mode.	113
StartLaserPowerAutoCalibration	Performs the laser power auto-calibration.	114
StopLaserPowerAutoCalibration	Stops the laser power auto-calibration.	114
StartLaserPowerMaintenance	Starts laser power maintenance.	114
StopLaserPowerMaintenance	Stops laser power maintenance.	114
GetLaserPowerMaintenanceOutput	Gets the laser power output.	115
SetDistancePointerStatus	In the Operation status, set the distance pointer to ON/OFF.	115
GetDistancePointerStatus	Gets the ON/OFF of the distance pointer in the operating status.	115
SetCumulativeMarkingCount	Sets the cumulative marking count.	116
GetCumulativeMarkingCount	Gets the cumulative marking count.	116
SetCumulativeMarkingCountString	Sets the cumulative marking count as a character string. (For VBA)	116
GetCumulativeMarkingCountString	Gets the cumulative marking count as a character string. (For VBA)	116
ApplyLaserScanningCalibration	Update the controller's correction status with the created LSC file.	117
GetOperatingTime	Gets the operation time of the controller.	117
GetScannerOperatingTime	Gets the operation time of the scanner.	117
GetShutterOperatingCount	Gets the number of shutter cycles.	117
GetLaserSafetyModuleOperatingCount	Gets the number of laser safety module cycles.	117
GetMarkingUnitTemperature	Gets the head temperature.	118
GetControllerTemperature	Gets the controller temperature.	118
GetLaserPowerCalibrationResult	Gets the results of the laser power calibration.	118
GetLaserAccumulatedTime	Gets the time that the laser has been in the excited state.	118
GetWindowCheckResult	Gets the final execution result of the lens inspection.	118
GetStartupWindowCheckEnabled	Gets whether or not to enable the lens inspection at startup.	118
SetStartupWindowCheckEnabled	Sets whether or not to enable the lens inspection at startup.	119
GetWindowCheckAlarmThreshold	Gets the warning threshold of the lens inspection.	119
SetWindowCheckAlarmThreshold	Sets the warning threshold of the lens inspection.	119
GetWindowCheckSensitivity	Gets the sensitivity adjustment of the lens inspection.	119
SetWindowCheckSensitivity	Sets the sensitivity adjustment of the lens inspection.	119
StartWindowCheck	Executes the lens inspection.	119

Property

LaserMarkerVersion	Gets the version number of the laser marker.	120
ControllerSerialNo	Gets the controller serial No.	120
HeadSerialNo	Gets the head serial No.	120

Class MBMatrix

Method

MatrixCell	Gets individual cell settings of a matrix.	121
SetWidth	Sets the width of a matrix.	121
SetHeight	Sets the height of a matrix.	121

Property

RowCount	Gets the number of rows in matrix.	122
ColumnCount	Gets the number of columns of a matrix.	122
X	Gets/sets the reference coordinate X of the matrix.	122
Y	Gets/sets the reference coordinate Y of the matrix.	122
CellWidth	Gets the width of one cell in a matrix.	122
CellHeight	Gets the height of one cell in a matrix.	122
MatrixWidth	Gets the width of the entire matrix.	123
MatrixHeight	Gets the height of the entire matrix.	123

Class MBMatrixCell

Property

OffsetX	Gets/sets the offset value of the X coordinate of an individual cell.	124
OffsetY	Gets/sets the offset value of the Y coordinate of an individual cell.	124
OffsetZ	Gets/sets the offset value of the Z coordinate of an individual cell.	124
OffsetAngle	Gets/sets the offset value of the θ angle of an individual cell.	125
IsMarkingEnable	Gets/sets whether marking is enabled for an individual cell.	125
MarkingDelayTime	Gets/sets the wait time for start marking of an individual cell.	125
OriginX	Gets the X coordinate of the point of origin in a cell.	125
OriginY	Gets the Y coordinate of the point of origin in a cell.	125

Class MBOperation

Method

SetCurrentCounterValue	Sets the current value of the counter.	126
GetCurrentCounterValue	Gets the current value of the counter.	126
SetCurrentCounterString	Sets the current value of the counter as a character string.	127
GetCurrentCounterString	Gets the current value of the counter as a character string.	127
SetCurrentCounterRepetition	Sets the repetition value of a counter.	128
GetCurrentCounterRepetition	Gets the repetition value of a counter.	128
SetCurrentCounterRepetitionString	Sets the current counter repetition marking count as a character string. (For VBA)	129
GetCurrentCounterRepetitionString	Gets the current counter repetition marking count as a character string. (For VBA)	129
GetJobTitle	Gets a program title.	130
GetWorkDistanceGap	Gets the working distance.	130
SetCharString2	Sets the character string.	130
GetCharString2	Gets the character string.	130
SetWorkpiecePositionAdjustment	Sets the workpiece position adjustment.	131
GetWorkpiecePositionAdjustment	Gets the workpiece position adjustment.	131
SetWorkpiecePositionAdjustmentZ	Sets the correction value in the height direction of the workpiece position adjustment.	132
GetWorkpiecePositionAdjustmentZ	Gets the correction value in the Z direction of workpiece position adjustment.	132
StartMarking	Starts marking.	132
StartGuideLaserMarking	Starts guide laser marking.	133
StopMarking	Stops marking.	133
GetCurrentJobNo	Gets the current program number running on the unit.	134
SetCurrentJobNo	Sets the program number to be run on the unit.	134
ClearError	Clears the error status in the unit.	134
BeginTransaction	Starts changing the mark setting of the controller.	135
CommitTransaction	Finishes changing the mark setting of the controller.	135
RollbackTransaction	Cancels changing the mark setting of the controller.	135
StartCameraScanningMode2	Starts camera scanning mode.	136

Interface List

FinishCameraScanningMode	Finish camera scanning mode.	136
SetCameraPosition	Sets the camera position.	137
OpenShutter	Opens the shutter.	137
CloseShutter	Closes the shutter.	137
SetMatrixCellOffset	Sets the offset of a matrix.	138
GetMatrixCellOffset	Gets the offset of a matrix.	138
SetMatrixCellMarkingEnable	Sets the marking flag of the matrix.	139
GetMatrixCellMarkingEnable	Gets the marking flag of the matrix.	139
Read2DCode2	Reads a 2D code.	140
Read2DCodePosition2	Reads a 2D code in a specified position.	141
Read2DCodeBlock	Reads a 2D code specified by block number and area.	141
Get2DCodeReadResult	Gets the reading result of the last 2D code.	141
GetCameraViewImage	Saves the displayed image on the finder view in Bitmap format.	142
StartZTrackingCalibration	Executes the calibration of the Z tracking.	142
CheckZPosition	Gets the Z coordinate of a desired point.	142
StartZTracking	Executes the Z tracking.	142
GetZTrackingResult	Gets the correction result of the Z tracking.	143
StartXYTracking	Executes the XY tracking.	143
GetXYTrackingResult	Gets the correction result of the XY tracking.	144
GetWindowCheckResult	Gets the result of the lens inspection before or after marking.	144
StartWidnwoMonitor	Executes the lens inspection before or after marking.	145
SetLighting	Sets the ON/OFF status of the lighting set by a job during operation.	145
GetLighting	Gets the ON/OFF status of the lighting set by a job during operation.	145

Property

IsCameraScanningMode	Gets whether the camera is being controlled.	146
IsExpanding	Gets whether expansion is in progress.	146
IsError	Gets whether an error has occurred.	146
IsWarning	Gets whether a warning is being issued.	146
IsReady	Gets the Ready status.	147
IsMarking	Gets whether marking is in progress.	147
IsShutterOpen	Gets the open/close status of the shutter.	147
IsLaserActivated	Gets the laser indicator (whether the laser excitation is supplied).	147
IoEncodedCharacter	Gets/sets the I/O encoded value.	148
IsCameraFinderView	Gets/sets whether to set the preview display mode to Finder.	148

Class MBEError

Property

ErrorCode	Gets the error code number.	149
ErrorCaption	Gets the error contents.	149

Class MBEErrorHistory

Property

ErrorCode	Gets the error codes registered in the error history.	150
ErrorCaption	Gets the error information of the errors registered in the error history.	150
ErrorDateTime	Gets the occurrence date and time of the errors registered in the error history.	150

Class MBControllerFile

● Property

FileName	Gets the file name.	151
JobNo	Gets the program No.	151

Class MBDxf

● Method

Convert	Convert the DXF file to a hatch logo file.	152
---------	--	-----

● Property

IsDetectPointElement	Sets whether to use the POINT element for computing the figure existing range.	152
----------------------	--	-----

Class MBOverprintingCode

● Property

CodeForm	Gets the code format (Normal/Black/White Inversion/Base).	153
LaserPower	Gets/sets the laser power of a multi-pass code.	153
IsLaserPowerBlockCommon	Gets/sets whether to use the common laser power parameter for blocks.	153
ScanSpeed	Gets/sets the scan speed of multi-pass code.	154
IsScanSpeedBlockCommon	Gets/sets whether to use the common scan speed parameter for blocks.	154
Frequency	Gets/sets the pulse frequency of multi-pass code.	154
IsFrequencyBlockCommon	Gets/sets whether to use the common pulse frequency parameter for blocks.	154
SpotVariable	Gets/sets the spot variable of multi-pass code.	155
IsSpotVariableBlockCommon	Gets/sets whether to use the common spot variable parameter for blocks.	155
Repetition	Gets/sets the repetition of multi-pass code.	155
IsRepetitionBlockCommon	Gets/sets whether to use the common repetition parameter for blocks.	155
FillLineInterval	Gets/sets the fill interval of multi-pass code.	156
IsFillLineIntervalBlockCommon	Gets/sets whether to use the common line interval parameter for blocks.	156
FillMargin	Gets/sets the shrink fill of a multi-pass code.	156
PatternBarcode	Gets/sets the pattern of a multi-pass barcode.	157
Pattern2D	Gets/sets the pattern of a multi-pass 2D code.	158

Class MB3DShape

● Property

X	Gets/sets the X coordinate of a 3D shape.	159
Y	Gets/sets the Y coordinate of a 3D shape.	159
Z	Gets/sets the Z coordinate of a 3D shape.	159

Class MBCommonBlock

● Property

LaserPower	Gets/sets the common laser power parameter for blocks.	160
ScanSpeed	Gets/sets the common scan speed parameter for blocks.	160
Frequency	Gets/sets the common pulse frequency parameter for blocks.	160
SpotVariable	Gets/sets the common spot variable parameter for blocks.	160
Repetition	Gets/sets the common repetition parameter for blocks.	161
QualityLevel	Gets/sets the common quality level parameter for blocks.	161
Z	Gets/sets the common Z coordinate parameter for blocks.	161
FillLineInterval	Gets/sets the common fill interval parameter for blocks.	161
SkipCross	Gets/sets the common skip cross parameter for blocks.	161

Class MBWindowCheck

● Property

Enabled	Gets/sets whether or not to enable the lens inspection before or after marking.	162
AlarmThreshold	Gets/sets the warning threshold of the lens inspection before or after marking.	162
Sensitivity	Gets/sets the sensitivity adjustment of the lens inspection before or after marking.	162

Class MBXYTracking

● Property

Enable	Gets/sets whether or not to enable the XY tracking.	163
CorrectionThreshold	Gets/sets the correlation value lower limit of the XY tracking.	163

Class MBZTracking

● Property

Enable	Gets/sets whether or not to enable the Z tracking.	163
--------	--	-----

● Method

GetHeightLimit	Gets the correction amount threshold settings of the Z tracking.	163
SetHeightLimit	Sets the correction amount threshold settings of the Z tracking.	163

Before Use

Marking Builder Plus ActiveX control allows users to easily implement various types of sophisticated processing performed with Marking Builder Plus. This ActiveX control dramatically reduces the amount of coding for editing mark data or managing the operation status of the laser marker.



“Marking BuilderPlus ActiveX control” is referred to as “MBPlusActiveX” hereafter.

Required devices and operating environment

This section describes the devices and operating environment required for using MBPlusActiveX.

Required devices

- for MD-X2000/2500 series
- USB cable or LAN cable
- The PC that Marking Builder Plus is installed on

Operating environment

Supported OS	Windows 7 (32/64 bit), Windows 8, 8.1 (32/64 bit) Windows 10 (32/64 bit)
Supported language	Japanese, English, Simplified Chinese, German, Korean, French, Spanish, Italian, Thai
CPU	2 GB or more (32 bit OS), 4 GB or more (64 bit OS)
Memory capacity	1 GHz or more
Display resolution	1024 * 768 pixels or more
HDD free space	1 GHz or more
Supported development language	Microsoft Visual Studio 2012-2019 Microsoft Excel 2010-2019 (VBA)



- Read carefully “Chapter 2 To Use Safely” in the User’s Manual of each model, and operate the device in the status to secure the safety using a safety function even when a problem like network failure occurs.
- Confirm that the device can be operated safely, considering the communication delay due to the load status of communication time or the network.



- Marking Builder Plus must be installed by a user with administrator privileges.
- To use the 3D portion of Marking Builder Plus, a graphics card supporting DirectX 9.0c or higher is required.
- Marking Builder Plus and an application incorporating MBPlusActiveX cannot communicate with a laser marker at the same time.
- We have checked the operation in the development language mentioned above; however, the operation cannot be guaranteed because it largely depends on the coding method.

Example of Each Development Language

The sample programs in Chapter 2 onwards are written in C#. Use an appropriate coding method according to the development language of the application. Note that we may not be able to reply to an inquiry on the procedure for incorporation or coding details of MBPlusActiveX. For further information, contact the distributors of the development languages.

About types

For the types in each development language environment, see the following.

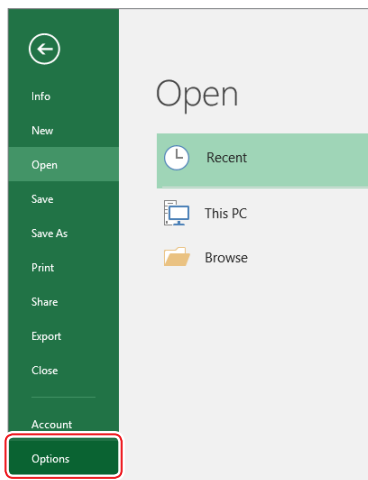
Notation in spec.	Detail	C# (managed)	VB.NET (managed)	C++/CLI (managed)	C++ (unmanaged)	VBA (unmanaged)	C++Builder (unmanaged)
byte	8bit	byte	Byte	byte	BYTE	Byte	BYTE
short	16bi	short	Short	short	short	Integer	short
int	32bit	int	Integer	int	long	Integer	long(int)
double	64bit	double	Double	double	double	double	double
out double	64 bit reference/pointer	out double	out double	out double	double *	out double	double *
string	Character string	System.String	System.String	System.String	BSTR	String	BSTR (WideString)
bool	True/False	bool	Boolean	Boolean	VARIANT_BOOL	Boolean	VARIANT_BOOL (bool)
IXxxx	Interface	IXxxx	IXxxx	IXxxx^	IXxxx	IXxxxPtr	IXxxx

How to refer to “MBPlusActiveX”

This section describes how to see “MBPlusActiveX” with Excel 2016 and Visual Studio 2019.

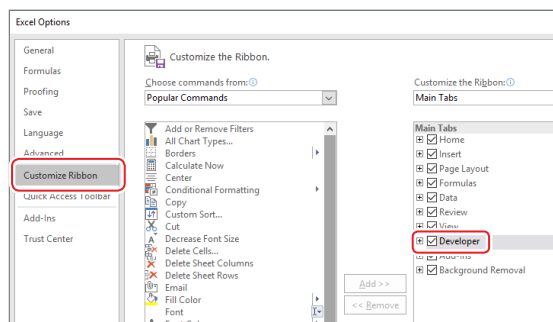
Excel 2016

1 Select [Options] from File menu.



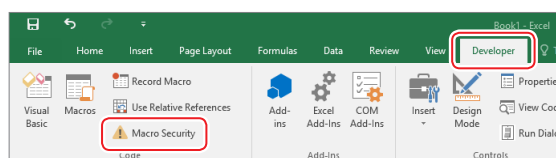
The “Excel Options” window appears.

2 In [Customize Ribbons], select [Developer] and then click “OK”.



The “Developer” tab is added to the menu bar.

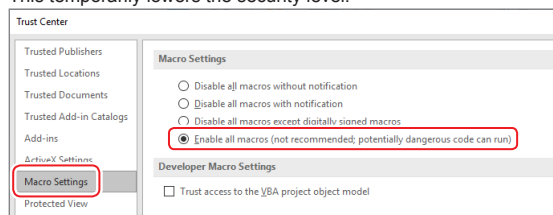
3 In the [Developer] tab, select [Macro Security].



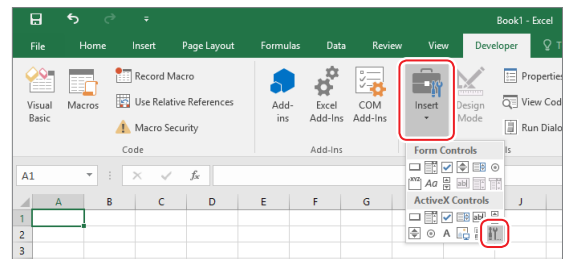
The Trust Center window is shown.

4 Select [Macro Settings], then [Enable all macros (not recommended; potentially dangerous code can run)], and click “OK”.

This temporarily lowers the security level.

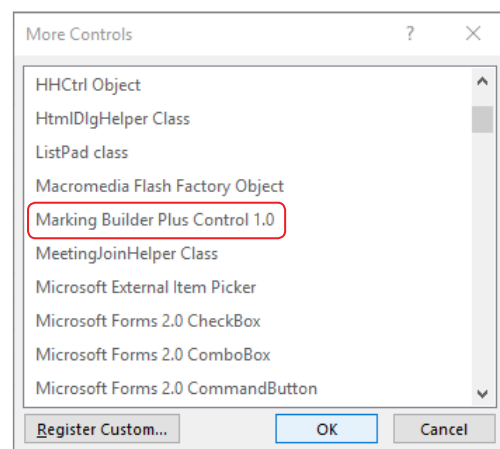


5 From the items in [Insert] in the Developer tab, select [More Controls].



The window for selecting a control is shown.

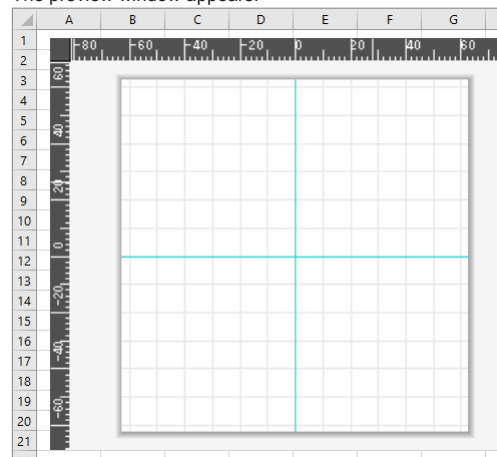
6 Select [Marking Builder Plus Control *.0] and click [OK].



The control selection window closes and the preview window for determining a size appears.

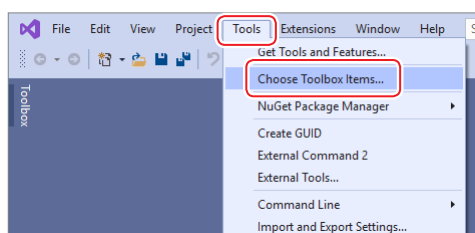
7 Select a rectangle of a necessary size.

The preview window appears.



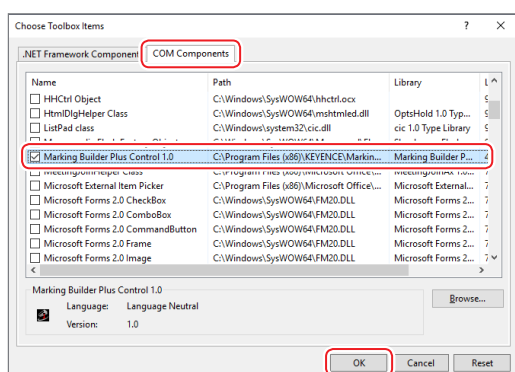
Visual Studio 2019

- 1 In the menu bar, select [Tools] - [Choose Toolbox Items].



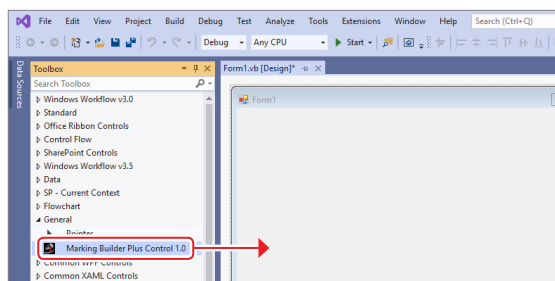
The window for selecting a toolbox item is shown.

- 2 In [COM Component], select [Marking Builder Plus Control *.0] and click [OK].

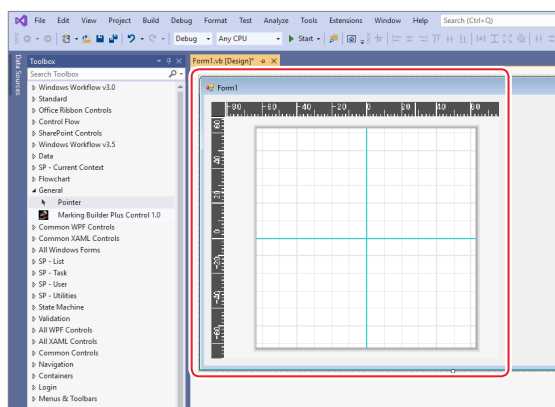


[Marking Builder Plus Control *.0] is added to the Toolbox.

- 3 Drag and drop [Marking Builder Plus Control *.0] from the Toolbox.



The preview window is shown,



Setting of the Target OS

MBPlusActiveX is the ActiveX control that runs in the Win32 environment.

The 64-bit version of Windows builds the target as a Win32 application. This built application must be run on WOW64.

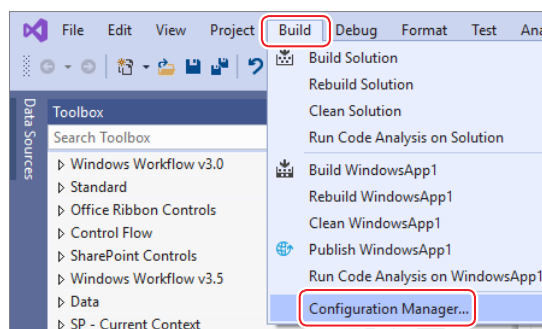
- Excel (2010 or later)

By default, Excel 2010 automatically selects the 32-bit version of Excel, but the 64-bit version of Excel can be selected and installed. MBActX runs only when the 32-bit version of Excel is installed.

- Visual Studio 2019

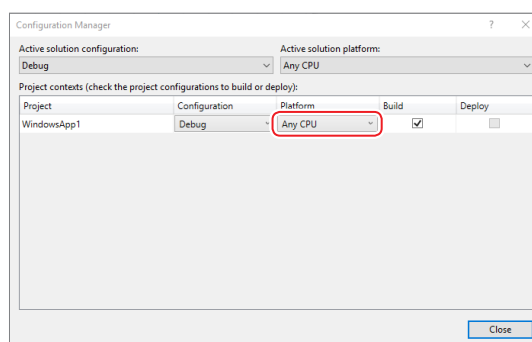
Applications that run on .NET Framework can be specified when the target OS is built.

- 1 Select [Configuration Manager] on [Build] in the menu bar.



The Configuration Manager window is shown.

- 2 Select the OS for the operation in [Platform].



Access to Interfaces in Development Environments

Excel 2016

● MBActX instance variable

The name "MBActX1" is set by default.

• Property

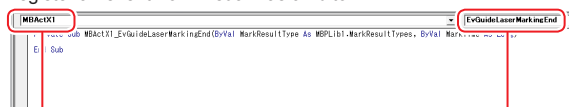
```
On Error GoTo ERRHANDLER
MBActX1.Block(2).LaserPower = 80
GoTo FINALLY
ERRHANDLER:
    MsgBox (Err.Description)
FINALLY:
```

• Method

```
On Error GoTo ERRHANDLER
MBActX1.Comm.Online
GoTo FINALLY
ERRHANDLER:
    MsgBox (Err.Description)
FINALLY:
```

• Event

Register an event from Visual Basic Editor.



Select MBActX1.

Events have been registered in the drop-down list. Selecting an event automatically generates code.

► Important

When a class other than MBActX is accessed, memory is dynamically reserved.
If a problem occurs with the automatic release by garbage collection, make sure to include the processing of releasing the memory after the use of it. (For details on the interface to reserve memory, see Chapter 2 onwards.)

```
On Error GoTo ERRHANDLER
Dim block As MBlock
Set block = MBActX1.Block(0) 'alloc memory
block.Text = "ABC"
GoTo FINALLY

ERRHANDLER:
    MsgBox (Err.Description)

FINALLY:
    'dealloc memory
    Set block = Nothing
```

Visual Studio 2019 (C#)

● Instance variable of the MBActX class

The name "axMBActX1" is set by default.

• Property

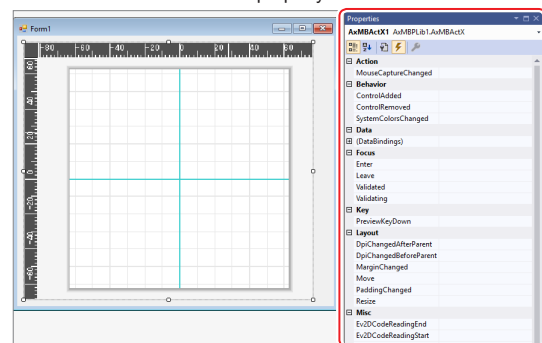
```
try
{
    axMBActX1.Block(2).LaserPower = 80;
}
catch (COMException error)
{
    MessageBox.Show(error.Message, Text);
}
```

• Method

```
try
{
    axMBActX1.InitMBActX(MBPLib1.MarkingUnitTypes.
        MARKINGUNIT_MDX2000);
}
catch (Exception, e)
{
    MessageBox.Show(error.Message, Text);
}
```

• Event

An event can be set from the property window.



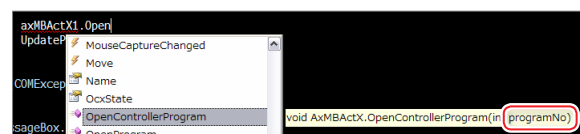
► Important

When a class other than MBActX is accessed, memory is dynamically reserved.
If a problem occurs with the automatic release by garbage collection, include the processing of releasing the memory at an appropriate timing. (For details on the interface to reserve memory, see Chapter 2 onwards.)

```
MBlock block = null;
try
{
    block = axMBActX1.Block(0); // alloc memory
    block.Text = "ABC";
}
catch (COMException err)
{
    MessageBox.Show(err.Message, Text);
}
finally
{
    // dealloc memory
    if (block != null)
    {
        Marshal.ReleaseComObject(block);
        block = null;
    }
}
```

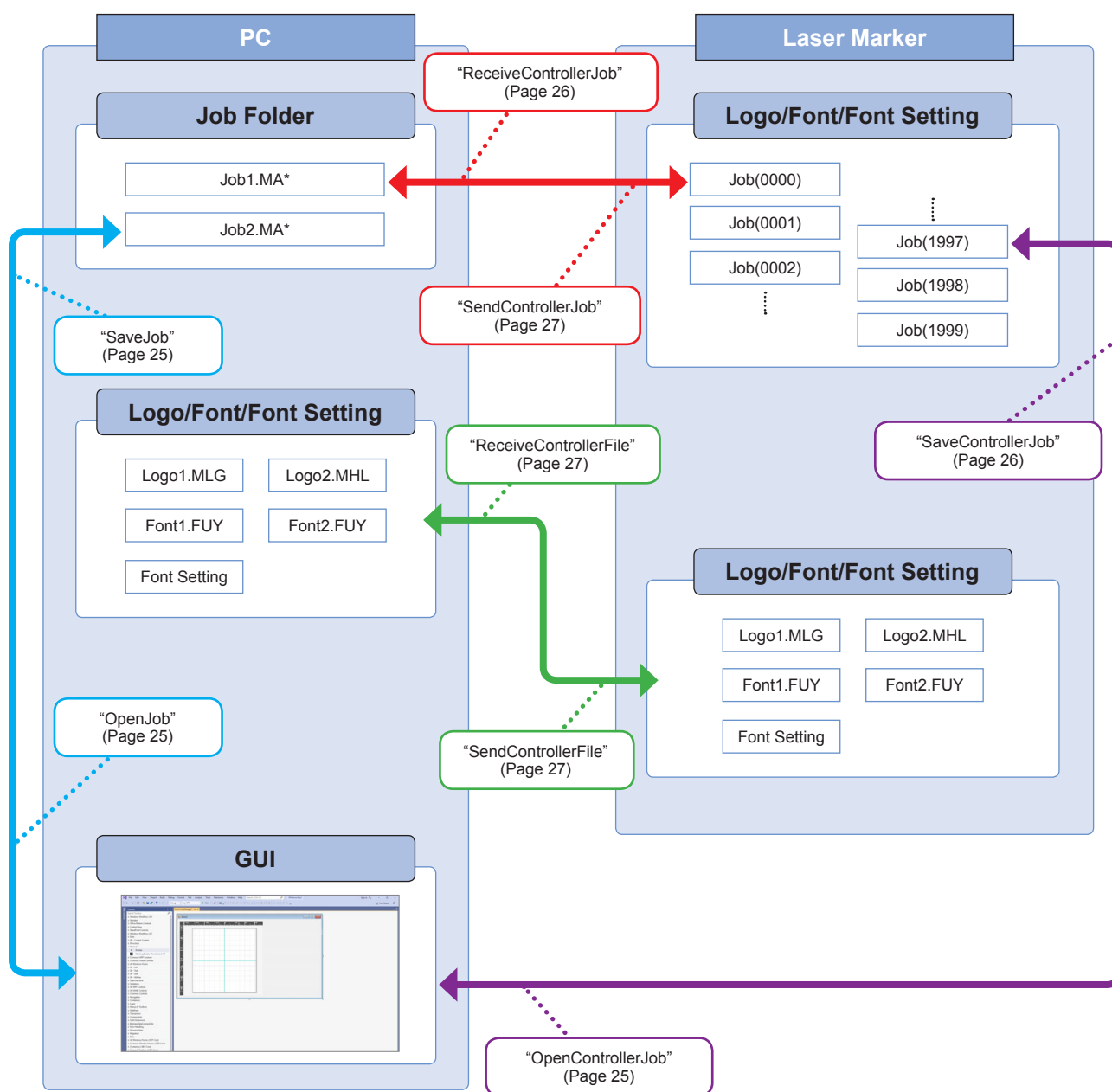
🔍 Point

Only for the functions provided by the MBActX class, an argument name begins with a lowercase character. The automatic generation processing of C# (processing of Visual Studio) automatically changes the argument of IDL (definition information) to lower case characters.



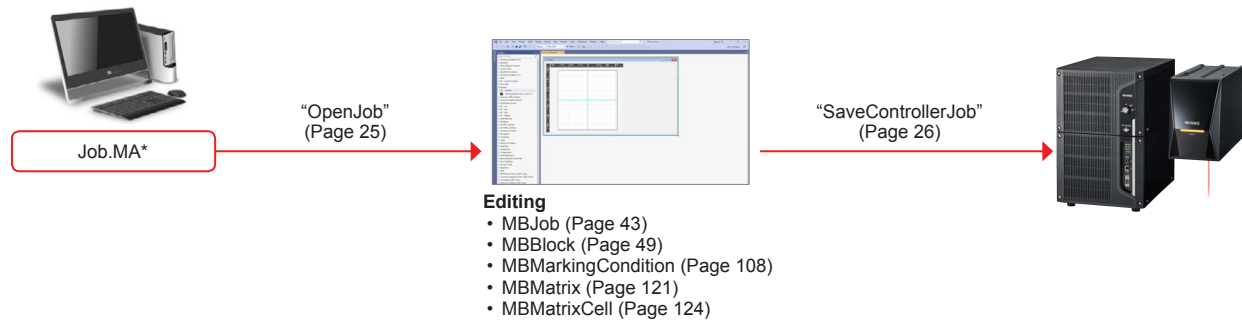
File operation from interface

In the laser marker file, there are "Job file (.MA*)", "Logo/hatch logo file (.MLG/.MHL)", "Font file (.FUY)" and "Font assignment information (FontSetting)". When operating a file from interface (GUI), the paths are shown below.



Procedures for opening job file inside a PC folder via GUI and transferring it to laser marker

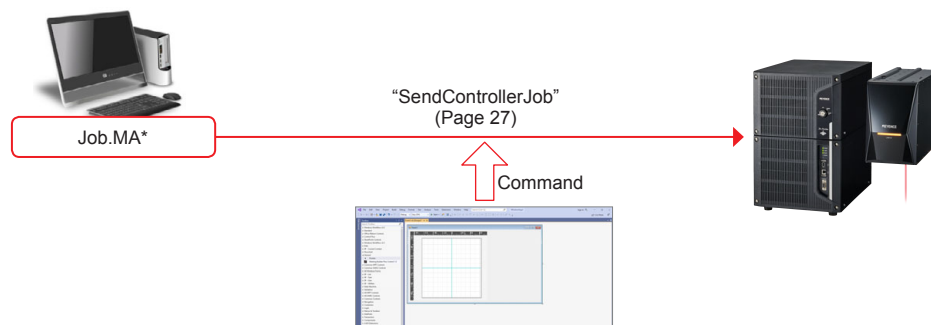
After opening the file via GUI, edit the file and transfer it.



Important In job file (.MA*file), since the logo, hatch logo, Workpiece image, and Z-MAP file which are used in the job are included, it is not necessary to transfer them via SendControllerFile separately.

Procedures for transferring a job file inside a PC folder directly to controller

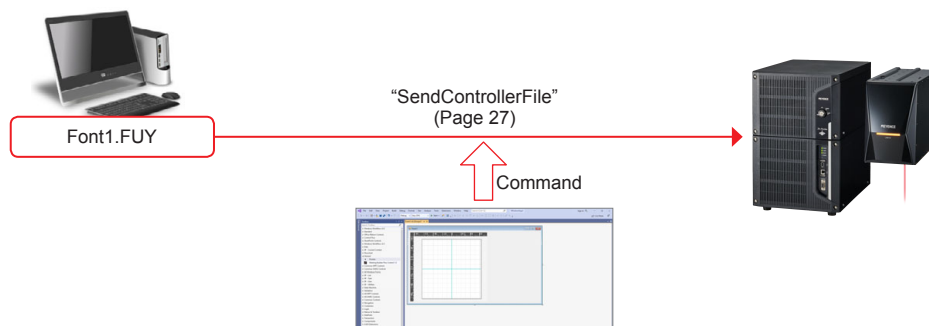
When transferring directly, although the job cannot be edited, the job file can be transferred through step 1.



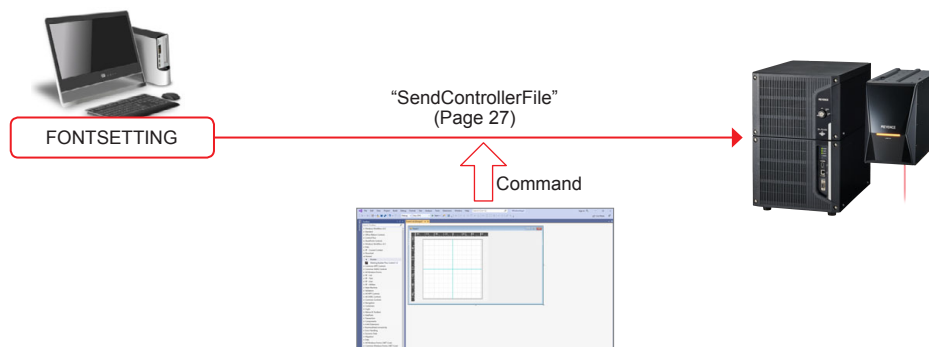
Procedures for transferring a new font file (.FUY)

When transferring a font file to the controller, the font file (.FUY) and font assignment information (Font Setting) are required to be transferred simultaneously.

1 Transfer a font file to the controller.



2 Transfer font assignment information to the controller.



- If the argument of SendControllerFile (Page 27) is set to "FONTSETTING", the assignment information on PC will be overwritten on laser marker.
- If assigning a font file (.FUY) to PC, use SetFontSetting (Page 29).
- If confirming the current assignment of the font file (.FUY), use GetFontSetting (Page 29).

Standard Programs

This section describes the flow in the case of a typical program.

Start of the application

1. Initial setting

MBActX.InitMBActEX (model, language)
Initializing MBActX and selecting model and language(Page 29)

MBActX.IsAutoRedraw ☐ True / False
Whether or not to update the preview automatically(Page 35)

MBActX.IsBlockingCommunication = True / False
Whether or not to wait until expansion/markings has finished before ending communication.(Page 35)

2. Online operation

● For USB connection

MBComm.ConnectionType
= ConnectionTypes.CONNECTION_USB
Whether or not to wait until expansion/markings has finished before ending communication.(Page 35)

→ One controller is connected to the PC (*Default)

MBComm.UsbControllerSerialNo = 0
Connect to one controller(Page 42)

→ Put a desired controller online among the controllers connected to the PC.

MBComm.UsbControllerSerialNo = #####
Specify the controller serial No.(Page 42)

MBComm.Online
Online.(Page 40)

● For Ethernet connection

MBComm.ConnectionType
= ConnectionTypes.CONNECTION_ETHERNET
Connection type setting(Page 41)

MBComm.IpAddress = ###.###.###.###
IP address setting(Page 41)

MBComm.Online
Online.(Page 40)

3. Setting editing operation

MBActX.Context = ContextTypes.CONTEXT_EDITING
Switch to editing mode(Page 33)

• Open a program from the PC memory

MBActX.OpenJob (file path)
Open a setting from the controller.(Page 25)

• Open a program from the controller memory

MBActX.OpenControllerJob (setting number)
Open a setting from the controller.(Page 25)

Editing

- Job information: MBJob (Page 43)
- Block information: MBBlock (Page 49)
- Marking conditions clipboards:
MBMarkingCondition (Page 108)
- Matrix information: MBMatrix (Page 121)
- Matrix cell information: MBMatrixCell (Page 124)

• Save to the PC

MBActX.SaveJob (file path)
Save the setting to the controller.(Page 25)

• Save to the controller

MBActX.SaveControllerJob (setting number)
Save the setting to the controller.(Page 26)



4. Controller operation

MBActX.Context =
ContextTypes.CONTEXT_CONTROLLER
Switch to operation mode (The preview window shows the
current program)
(Page 33)

● When switching a setting number

MBOperation.SetCurrentJobNo (setting number)
Switch the setting number (Page 134)

● For fine adjustment

MBOperation.BeginTransaction()
Start of fine adjustment (Page 135)



Editing for the current setting
• Marking flag
• Changing character string
etc.



MBOperation.CommitTransaction()
Completion of fine adjustment (Page 135)

● When capturing by built-in camera

MBOperation.IsCameraFinderView = TRUE
Display the finder image on the preview window (Page 148)



MBOperation.StartCameraScanningMode2()
Start finder mode (Page 136)



GetCameraViewImage()
Save the finder view image
(Page 142)



MBOperation.SetCameraPosition()
Move the finder coordinates (Page 137)

● When using a DXF file

Dxf.Convert (file name, hatch logo name, whether to display
progress)
Convert a DXF file to a logo file (Page 152)



MBActX.SendControllerFile (hatch logo file name. MHL)
Transfer the logo file to the controller (Page 27)



MBOperation.StartMarking ()
Start marking (Page 132)

● When reading a 2D code

MBOperation.Read2DCodePosition2()
Read the 2D code at the specified position (Page 141)



MBOperation.Get2DCodeReadResult()
Get the reading result of the last 2D code (Page 141)

Other Notes

- If an exception occurs, the original value is retained.
- About the parameter (out double) of floating point representation

E.g.) For the block X (the significant figures are the thousandths-place digit), the input value "10.4567" is set as "10.457".
The algorithm of rounding adds 0.5 to the fraction and cuts off the lower-order digits regardless of the target sign.

- The property that is enabled or disabled depending on the parameter of another property must be accessed only when it is enabled.

- Changing a parameter unrelated to drawing does not update the drawing.

E.g.) LaserPower

- Alphabetical characters used in a file name or an extension are not case-sensitive.

When overwriting is done with loading etc., the overwriting will be made to the original file name/extension.

E.g.) When the controller contains "ABC.FUY":

For the ReceiveControllerFile argument, both "ABC.FUY" and "abc.FUY" are available.

E.g.) When a font file named "abc.FUY" in "C:\Users\Username\Documents\Marking Builder Plus\Font" exists, if "ABC.FUY" is read from the controller, the file remains under the name, "abc.FUY".

- If multiple events occur in one method, the following occurrence order is guaranteed:

- EvExpansionStart -> EvExpansionEnd
- EvMarkingStart2 -> EvMarkingEnd2

However, if combined with other events than the above, the occurrence order may differ.

E.g.)When MBEquipment::StopLaserPowerAutoCalibration() is executed, the following three patterns of events may occur:

- EvError -> EvExpansionStart -> EvExpansionEnd
- EvExpansionStart -> EvError -> EvExpansionEnd
- EvExpansionStart -> EvExpansionEnd -> EvError

- Incorporate the program via VBA, and if the argument is enclosed in (), an error may occur. In such case, put "Call" at the beginning.

MBActX

How to Read Interface Table

(1)	Type	Method	Mode	Operation	✓
(2)	Memory allocation	-		Editing	✓
(3)	Event	-	Blocking communication	-	-
	Argument	-			
	Return value	bool	True False	Successful Failed	
	Note				
	Detailed example C#	if(axMBActX1.ZoomAdjust()){ //TODO Write the processing to be performed if the condition is true. }			
	Simple example VBA	MBActX1.ZoomAdjust			

(3) Event

EvExpansionStart	Occurs when expansion has started. (Page 36)
EvExpansionEnd	Occurs when expansion has been completed. (Page 36)
EvMarkingStart2	Occurs when marking has started. (Page 36)
EvMarkingEnd2	Occurs when marking has been completed. (Page 36)
EvGuideLaserMarkingStart	Occurs when guide laser marking has started. (Page 37)
EvGuideLaserMarkingEnd	Occurs when guide laser marking has been completed. (Page 37)
EvConnectionError	Occurs when the physical connection of USB or Ethernet is disconnected. (Page 37)
EvError	Occurs when the error status has changed. (Page 38)
Ev2DCodeReadingStart	Occurs when 2D code reading has started. (Page 38)
Ev2DCodeReadingEnd	Occurs when 2D code reading has been completed. (Page 38)
EvMarkingConfirmationStart	Occurs when marking verification has started. (Page 39)
EvMarkingConfirmationEnd	Occurs when marking verification has been completed. (Page 39)
EvMarkingWorkflowEnd	It will occur when the set operation before marking, marking, and operation after marking are completed. (The check of marking energy is out of target.) (Page 39)

(1) Type

Method	Operation for an object
Property(R)	Object in the read only status
Property (RW)	Object in the writable status
Event	Event that may occur at this interface

(2) Memory allocation

When this item is marked with ✓, memory is dynamically allocated to generate an instance.



If a problem occurs with automatic release by garbage collection, perform release processing after the object accessing is completed.

```

MBBlock block = null;
try
{
    block = axMBActX1.Block(0); // alloc memory
    block.Text = "ABC";
}
catch (COMException err)
{
    MessageBox.Show(err.Message, Text);
}
finally
{
    // dealloc memory
    if (block != null)
    {
        Marshal.ReleaseComObject(block);
        block = null;
    }
}

```

(4) Mode

- In the case of Context = Editing(Page 33)
It will turn to the editing mode.
If "Editing" is marked with "✓", the interface is available.
- In the case of Context = Controller(Page 33)
It will turn to the operation mode. The current settings will appear on the preview window.
If "Operation" is marked with "✓", the interface is available.

(5) Blocking communication (IsBlocking Communication)

When this item is marked with "✓", the function exits in the case of non-blocking communication without waiting for the processing to be completed.



You can select a communication method from either Blocking (synchronous) communication or Non-blocking (asynchronous) communication.

- When IsBlocking Communication = True (Page 35), blocking communication is enabled. When executing a method including expansion processing or marking processing, the method does not exit until the expansion/ marking is finished. Although program implementation will be easier, parallel processing may be difficult.
- When IsBlocking Communication = False (Page 35), non-blocking communication is enabled. When executing a method including expansion processing or marking processing, the method exits before the expansion/ marking is finished. Parallel processing is possible, but an event handler must be securely designed.

Method

ZoomAdjust

Adjusts the preview window to show all blocks.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	bool	True False	Successful Failed	
Note				
Detailed example C#	<pre>if(axMBActX1.ZoomAdjust()){ //TODO Write the processing to be performed if the condition is true. }</pre>			
Simple example VBA	MBActX1.ZoomAdjust			

ZoomAll

Displays the entire marking area in the preview window.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	bool	True False	Successful Failed	
Note	Can be performed even when there is no content in the marking field.			
Detailed example C#	<pre>if(axMBActX1.ZoomAll()){ //TODO Write the processing to be performed if the condition is true. }</pre>			
Simple example VBA	MBActX1.ZoomAdjust			

Zoom

Enlarges the preview window at the specified coordinate and scale ratio.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	double	CenterPointX	Central X coordinate	3rd decimal place
	double	CenterPointX	Central Y coordinate	3rd decimal place
	int	Zoom	Scale ratio (%) 25 - 40000	
Return value	bool	True False	Successful Failed	
Note	It is successful regardless of being drawing or not. Exception: If out of input range The "Out of range error" occurs.			
Detailed example C#	<pre>if(axMBActX1.Zoom(-2.500, 10.000, 120)){ //TODO Write the processing to be performed if the condition is true. }</pre>			
Simple example VBA	MBActX1.Zoom 10, 10, 500			

ZoomAdjustMatrixCell

Displays the preview adjusted to the specified cell.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	Reserved	Reserved (Fixed to 0)	
	int	Row	Row	
	int	Column	Column	
Return value	-			
Note	Exception: If the number of rows/columns is out of input range The out of input range error occurs. If no matrix exists An "unsupported error" occurs.			
Detailed example C#	<pre>int row = 2; int column = 3; try{ if(axMBActX1.Matrix(0) != null){ axMBActX1.ZoomAdjustMatrixCell(0,row,column); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.ZoomAdjustMatrixCell 0, 2, 3			

ShowBlockHighlighting

Highlights a block in the preview window.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	BlockNo	Block number	
Return value	-			
Note	If non-existent block is selected, no blocks will be highlighted. If a block is currently selected, the new block will be selected instead. (Multiple blocks cannot be selected.)			
Detailed example C#	axMBActX1.ShowBlockHighlighting(0);			
Simple example VBA	MBActX1.ShowBlockHighlighting 0			

HideBlockHighlighting

Cancels the highlight in the preview window.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	-			
Note	No change occurs when this method is executed without a block highlighted.			
Detailed example C#	axMBActX1.HideBlockHighlighting();			
Simple example VBA	MBActX1.HideBlockHighlighting			

Redraw

Re-draws the preview.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	-			
Note	Use this method when IsAutoRedraw = false (update the drawing automatically OFF). The preview in the display area is updated. When IsAutoRedraw = true, this method is unnecessary because the drawing in the display area is automatically updated.			
Detailed example C#	axMBActX1.Redraw();			
Simple example VBA	MBActX1.Context = CONTEXT_EDITING MBActX1.OpenControllerJob 0 MBActX1.IsAutoRedraw = False 'Auto update OFF MBActX1.Block(0).Text = "Test" 'Change character string MBActX1.Redraw 'The drawing is updated at this timing			

OpenJob

Opens a job file in the PC for editing.

Type	Method	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	string	FileName	File name	
Return value	bool	True False	Successful Failed *1	
Note	Specify the path where the target file is saved. When an absolute path is specified, the file in the specified path is opened. To open a file in the job data folder, you can also open it by specifying the file name only. Note that if the FileName argument is a null character string (""), the Open dialog starts and you can specify a job file. If an image is displayed on the finder by "StartCameraScanningMode", the finder display is ended. Exception: If a non-existent job file is specified The message, "No-File Error", occurs. If a file without its extension is specified The message, "Failed to open transmission file", occurs. * 1. When clicking "Cancel" on the Open dialog, "false" is returned.			
Detailed example C#	<pre>try{ if(axMBActX1.OpenJob("MarkingJobName.MA1")){ //TODO Write the processing to be performed if the condition is true. } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.OpenJob "TEST.MA1"			

SaveJob

Saves the job file being edited to the PC.

Type	Method	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	string	FileName	File name	
Return value	bool	True False	Successful Failed	
Note	Specify the path for the settings to be saved. When an absolute path is specified, the target file is saved to the specified path. When a file name only is specified, the file is saved to the job data folder. Note that if the FileName argument is a null character string (""), the Open dialog starts and you can specify a job file. Exception: If no folder exists DirectoryNotFoundException occurs. If a file without its extension is specified The message, "Failed to open transmission file", occurs.			
Detailed example C#	<pre>try{ if(axMBActX1.SaveJob("MarkingJobName.MA1")){ //TODO Write the processing to be performed if the condition is true. } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.SaveJob "TEST.MA1"			

OpenControllerJob

Opens a job saved in the controller for editing.

Type	Method	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	JobNo	Setting number	
Return value	-			
Note	If Argument is a numerical value, the program No. in the unit is read. If an image is displayed on the finder by "StartCameraScanningMode", the finder display is ended. Exception: If a program is read to the unit not connected The "Communication error" occurs. If job No. without setting data is specified The message, "Failed to open transmission file", occurs. If the job No. is out of the range when a job is read to the unit The message, "Job No. range error", occurs.			
Detailed example C#	try{ axMBActX1.OpenControllerJob(3); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			
Simple example VBA	MBActX1.OpenControllerJob 0			

SaveControllerJob

Saves the job being edited to the controller.

Type	Method	Mode	Operation	-
Memory allocation	-			Editing
Event	EvExpansionStart/ End	Blocking communication	✓	
Argument	int	JobNo	Setting number	
Return value	-			
Note	Exception: If a program is read to the unit not connected The "Communication error" occurs. If job No. without setting data is specified The message, "Job No. unregistration error", occurs. If the job No. is out of the range when a job is read to the unit The message, "Job No. range error", occurs.			
Detailed example C#	<pre>try{ axMBActX1.SaveControllerJob(3); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.SaveControllerJob 0			

DeleteControllerJob

Deletes a job from the controller.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	JobNo	Setting number	
Return value	-			
Note	Exception: If not connected to the unit The "Communication error" occurs. If job No. without setting data is specified The message, "Job No. unregistration error", occurs. If the job No. is out of the range when a job is read to the unit The message, "Job No. range error", occurs. When the current job is specified The "The running job operation error" occurs.			
Detailed example C#	try{ axMBActX1.DeleteControllerJob(3); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			
Simple example VBA	MBActX1.DeleteControllerJob 0			

ReceiveControllerJob

Copies and saves a job in the controller to the PC.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	string	FileName	File name	
	int	JobNo	Setting number	
Return value	bool	True	Successful	
		False	Failed	
	-			
Note	Specify the path for the settings to be saved. When an absolute path is specified, the target file is saved to the specified path. When a file name only is specified, the file is saved to the job data folder. Note that if the FileName argument is a null character string (""), the Open dialog starts and you can specify a job file. Exception: If not connected to the unit The "Communication error" occurs. If a file without its extension is specified The message, "Failed to open transmission file", occurs. If job No. without setting data is specified The message, "Job No. unregistration error", occurs. If the job No. is out of the range when a job is read to the unit The message, "Job No. range error", occurs.			
Detailed example C#	<pre>try{ if (axMBActX1.ReceiveControllerJob ("MarkingJob.MA1", 3)) { //TODO Write the processing to be performed if the condition is true. } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.ReceiveControllerJob "TEST.MA1", 0			

SendControllerJob

Copies and saves the job file in the PC to the controller.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvExpansionStart/ End	Blocking communication	✓	
Argument	string	FileName	File name	
	int	JobNo	Setting number	
Return value	bool	True	Successful	
		False	Failed	
		-		
Note	Send the file in the specified path. When an absolute path is specified, the target file in the specified path is sent. When the file name only is specified, open the file in the job data folder. Note that if the FileName argument is a null character string (""), the Open dialog starts and you can specify a job file. Exception: If not connected to the unit The "Communication error" occurs. If a file without its extension is specified An "unsupported error" occurs. If job No. without setting data is specified The message, "Job No. unregistration error", occurs. If the job No. is out of the range when a job is read to the unit The message, "Job No. range error", occurs.			
Detailed example C#	<pre>try{ if (axMBActX1.SendControllerJob (3, "MarkingJobName.MA1")) { //TODO Write the processing to be performed if the condition is true. } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.SendControllerJob 0, "TEST.MA1"			

DeleteControllerFile

Deletes a file from the controller.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	string	ControllerFileName	File name	
Return value	-			
Note	Exception: If not connected to the unit The "Communication error" occurs. If a file without its extension is specified The message, "Failed to open transmission file", occurs. If the extension does not consists of only upper- or lower-case characters (when mixed) The message, "Failed to open transmission file", occurs.			
Detailed example C#	<pre>try{ axMBActX1.DeleteControllerFile("LogoFileName.MLG"); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.DeleteControllerFile "TEST.MLG"			

ReceiveControllerFile

Copies and saves a file in the controller to the PC.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	string	ControllerFileName	File name	
Return value	-			
Note	Exception: If not connected to the unit The "Communication error" occurs. If a file without its extension is specified The message, "Failed to open transmission file", occurs. If the extension does not consists of only upper- or lower-case characters (when mixed) The message, "Failed to open transmission file", occurs.			
Detailed example C#	<pre>try{ axMBActX1.ReceiveControllerFile("LogoFileName.MLG"); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.ReceiveControllerFile "TEST.MLG"			

SendControllerFile

Copies and saves the target file in the PC to the controller.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvExpansionStart/ End	Blocking communication	✓	
Argument	string	FileName	Logo/font	File name
			Font assignment	"FONTSETTING"
Return value	-			
Note	Specify the absolute paths of the logo and font file to transfer. For font assignment, the internal data of the system is transferred. For ArgumentFileName, specify "FONTSETTING". Exception: If not connected to the unit The "Communication error" occurs. If a file without its extension is specified The message, "Failed to open transmission file", occurs. If the extension does not consists of only upper- or lower-case characters (when mixed) The message, "Failed to open transmission file", occurs.			
Detailed example C#	<pre>try{ axMBActX1.SendControllerFile("LogoFileName.MLG"); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Assign Arial font in the computer to font No.2 of controller MBActX1.SendControllerFile "C:\Users\[UserName]\Documents\Marking Builder Plus\Sysfont\Arial.FUY" MBActX1.SetFontSetting 2, "C:\Users\[UserName]\Documents\Marking Builder Plus\Sysfont\Arial.FUY" MBActX1.SendControllerFile "FONTSETTING"			

ReceiveControllerBufferFile

Copies and saves a file in the controller's logo custom character buffer to the PC.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	string	FileName	File name	
Return value	-			

Note

When data is transferred to "Logo/custom character buffer" by SendControllerBufferFile, the data can be read by ReceiveControllerBufferFile.
For details on how to use this method, see SendControllerBufferFile.
Exception:
If not connected to the unit
The "Communication error" occurs.
If a file without its extension is specified
The message, "Failed to open transmission file", occurs.

Detailed example C#

```
try{
    axMBActX1.ReceiveControllerBufferFile("LogoFileName.MLG");
}catch (System.Runtime.InteropServices.COMException error){
    MessageBox.Show(error.Message);
}
```

Simple example VBA

MBActX1.ReceiveControllerBufferFile "Users\[UserName]\Documents\Marking Builder Plus\Sysfont\TEST.MLG"

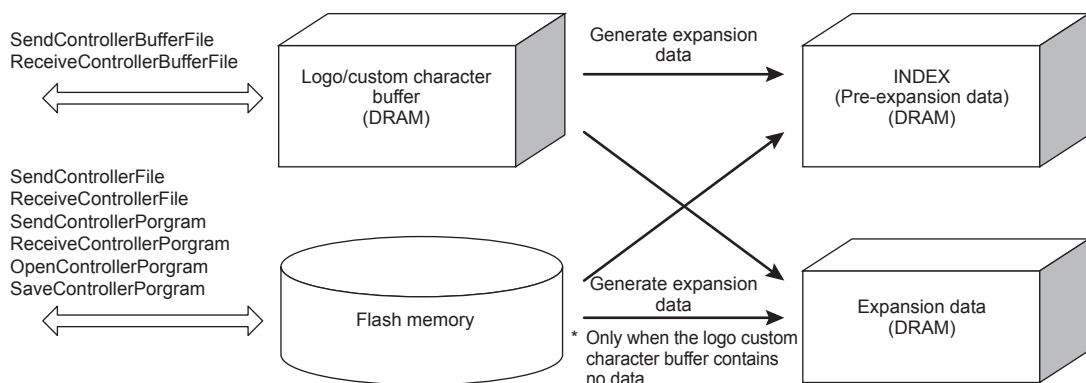
SendControllerBufferFile

Directly expands a file in the PC to the controller's logo custom character buffer.

Type	Method	Mode	Operation	✓	Event	EvExpansionStart/End		
Memory allocation	-		Editing	✓	Argument	string	FileName	Name of logo/hatch file in PC
Blocking communication	✓				Return value	-		

Note

To overwrite the logo (hatch logo) frequently, use this method instead of SendControllerFile.
Directly updates the data in the expansion memory without updating the file in the built-in flash memory.
Specify the absolute path of the logo (hatch logo) file.



Restrictions

- Files are just allocated on the DRAM, so they do not remain when the power is turned off.
- The data saved in the logo custom character buffer is not a target when a backup is made with Marking Builder Plus.
- There will be inconsistency between a file in the flash memory and that in the buffer.
- The data in the logo/custom character buffer is prioritized for actual marking.
- If the logo used for mark settings does not exist in the logo/custom character buffer or the flash memory, a logo file error occurs. This error may occur when the program No. is switched or the system is started.
- The list of the files in the logo/custom character buffer cannot be obtained from Marking Builder Plus or the console.
- When using Marking Builder Plus, the console, or MBPlusActiveX (OpenControllerJob/ReceiveControllerJob), obtains only logos saved in the flash memory.

When data is transferred to "Logo/custom character buffer" by SendControllerBufferFile, the data can be read by ReceiveControllerBufferFile.

- The Operation monitor in Marking Builder Plus and the Operation monitor of MBPlusActiveX displays a preview with the logo (hatch logo) saved in the flash memory.

Exception:

If not connected to the unit
The "Communication error" occurs.
If a file without its extension is specified
The message, "Failed to open transmission file", occurs.

Detailed example C#

```
try{
    axMBActX1.SendControllerBufferFile("LogoFileName.MLG");
}catch (System.Runtime.InteropServices.COMException error){
    MessageBox.Show(error.Message);
}
```

Simple example VBA

MBActX1.SendControllerBufferFile "Users\[UserName]\Documents\Marking Builder Plus\Logo\TEST.MLG"

GetFontSetting

Gets the font assignment information from the PC.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	FontNo	Font No.	
Return value	string		Font name	
Note	If the return value is a space character, font assignment has not been set. Exception: If the font No. is out of input range The out of input range error occurs.			
Detailed example C#	<pre>try{ string FontName = axMBActX1.GetFontSetting(2); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>Dim A As String A = MBActX1.GetFontSetting(0)</pre>			

SetFontSetting

Sets the font assignment information in the PC.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	FontNo	Font No.	
	string	FontFilePath	FUY file path	
Return value	-			
Note	To delete the font from the font assignment, specify the null character, "". Because fonts No. -1, 0, and 1 are the fixed fonts, they cannot be set. Exception: If the font No. is out of input range The out of input range error occurs.			
Detailed example C#	<pre>try{ axMBActX1.SetFontSetting(2, "FontName.FUY"); // set axMBActX1.SetFontSetting(3, ""); // release }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	See the example of SendControllerFile.			

InitMBActX

Initializes ActiveX and selects a connection destination.

Type	Method	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	MarkingUnitTypes	MarkingUnitType	MARKINGUNIT_MDX2000	
			MARKINGUNIT_MDX2020	
			MARKINGUNIT_MDX2050	
			MARKINGUNIT_MDX2500	
			MARKINGUNIT_MDX2520	
			MARKINGUNIT_MDX400	
			MARKINGUNIT_MDX450	
Return value	-			
Note	This method must be called at the beginning of the program to initialize MBActX. The model cannot be changed online, so change it offline. The language for ActiveX is automatically set depending on the language for the OS. Exception: If online A communication error occurs.			
Detailed example C#	<pre>try{ axMBActX1.InitMBActX(MarkingUnitTypes.MARKINGUNIT_MDX2000); } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.InitMBActX(MARKINGUNIT_MDX2000)			

InitMBActXEx

Initializes ActiveX in the specified language and selects a connection destination.

Type	Method	Mode	Operation	-
Memory allocation	-	Blocking communication	Editing	✓
Event	-		-	
Argument	MarkingUnitTypes	MarkingUnitType	MARKINGUNIT_MDX2000	MARKINGUNIT_MDX2520
			MARKINGUNIT_MDX2020	MARKINGUNIT_MDX400
			MARKINGUNIT_MDX2050	MARKINGUNIT_MDX450
			MARKINGUNIT_MDX2500	
	LanguageTypes	LanguageType	LANG_DE //German	LANG_FR //French
			LANG_EN //English	LANG_ESMX //Spanish
			LANG_JA //Japanese	LANG_IT //Italian
			LANG_KO //Korean	LANG_TH //Thai
			LANG_ZHCN //Simplified Chinese	
Return value	-			
Note	This method must be called at the beginning of the job to initialize MBActX. The model cannot be changed online, so change it offline. Exception: If online A communication error occurs.			
Detailed example C#	<pre>try{ axMBActX1.InitMBActXEx(MarkingUnitTypes.MARKINGUNIT_MDX2000, LanguageTypes.LANG_JA); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.InitMBActXEX(MARKINGUNIT_MDX2000,LANG_JA)			

Block

Gets the block information.

Type	Method	Mode	Operation	-
Memory allocation	✓			Editing
Event	-	Blocking communication	-	
Argument	int	BlockNo	Block number	
Return value	MBBlock		Block information object	
Note	Exception: If out of input range The out of input range error occurs.			
Example	See the MBBLOCK object.			

Matrix

Gets the matrix object.

Type	Method	Mode	Operation	-
Memory allocation	✓		Editing	✓
Event	-	Blocking communication	-	
Argument	int	Reserved	Reserved (Fixed to 0)	
Return value	MBMatrix	-	Matrix object	
Note	If no matrix exists, null is returned.			
Detailed example C#	<pre>try{ if(axMBActX1.Matrix(0) != null) }</pre>			
Simple example VBA	MBActX1.Context = CONTEXT_EDITING MBActX1.OpenJob ("test.MA1") If Not MBActX1.Matrix(0) Is Nothing Then MBActX1.Matrix(0).X = -10 Else MsgBox ("There is not matrix") End If MBActX1.SaveJob ("test.MA1")			

Shape

Gets the 3D shape setting object.

Type	Method	Mode	Operation	-
Memory allocation	✓		Editing	✓
Event	-	Blocking communication	-	
Argument	int	ShapeNo	3D shape setting number	
Return value	MB3DShape	-	3D shape setting object	
Note	If 3D shape setting for the specified No. does not exist, null is returned.			
Detailed example C#	<pre>int shapeNo = 0; try{ if(axMBActX1.Shape(shapeNo) != null){ axMBActX1.Shape(shapeNo).X = 10.000; axMBActX1.Shape(shapeNo).Y = 20.000; axMBActX1.Shape(shapeNo).Z = -3.000; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	If Not MBActX1.Shape(0) Is Nothing Then MBActX1.Context = CONTEXT_EDITING MBActX1.OpenJob ("test.MA1") If Not MBActX1.Shape(0) Is Nothing Then MBActX1.Shape(0).X = 10 Else MsgBox ("There is not 3D shape") End If MBActX1.SaveJob ("test.MA1")			

ControllerFile

Gets the file object of the controller.

Type	Method	Mode	Operation	✓
Memory allocation	✓			Editing
Event	-	Blocking communication	-	
Argument	int	fileIndex	File index	
Return value	MBControllerFile		Controller file information	
Note	Exception: If out of input range The out of input range error occurs.			
Example	See the MBControllerFile object.			

ErrorHistory

Gets the error history.

Type	Method	Mode	Operation	-
Memory allocation	✓			Editing
Event	-	Blocking communication	-	
Argument	int	ErrorHistoryIndex	Error history index	
Return value	MErrorHistory		Error history	
Note	Exception: If out of input range The out of input range error occurs.			
Example	See the MErrorHistory object.			

MarkingCondition

Gets the marking parameters clipboard object.

Type	Property(R)	Mode	Operation	-
Memory allocation	✓		Editing	✓
Event	-	Blocking communication	-	
Argument	int	MarkingConditionNo	Marking parameters clipboard object No.	
Return value	MBMarkingCondition		Marking parameters clipboard object	
Note	-			
Example	See the MBMarkingCondition object.			

UpdateErrorCount

Gets the number of current errors.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int		Error count in the history	
Note	Update the error information in the cache.			
Example	See the MError object.			

UpdateErrorHistoryCount

Gets the total number of errors listed in the error history.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int		Error count in the history	
Note	Update the error history information in the cache.			
Example	See the MBErrHistory object.			

Error

Gets the list of current errors.

Type	Method	Mode	Operation	✓
Memory allocation	✓			Editing
Event	-	Blocking communication	-	
Argument	int	ErrorIndex	Error index	
Return value	MError		Error list	
Note	Exception: If out of input range The out of input range error occurs.			
Example	See the MErrorHistory object.			

UpdateControllerFileCount

Gets the number of files in the controller.

Type	Method	Mode	Operation	✓
Memory allocation	-			Editing
Event	-	Blocking communication	-	
Type	ControllerFileTypes	Controller FileType	CONTROLLERFILE_CHARACTER //Custom character file on the controller side	
			CONTROLLERFILE_FONT //System font	
			CONTROLLERFILE_LOGO //Logo	
			CONTROLLERFILE_JOB //Job on the controller side	
			CONTROLLERFILE_ZMAP //ZMAP file on the controller side	
Note	Cache the file name at the same time. The file name can be obtained from the MBControllerFile object.			
Example	See the MBControllerFile object.			

GetWindowCheck

Gets the window maintenance information object.

Type	Method	Mode	Operation	-
Memory allocation	✓		Editing	✓
Event	-	Blocking communication	-	
Argument	WindowCheck TimingTypes	timingType	WINDOW_CHECK_BEFORE_MARKING //Before marking WINDOW_CHECK_AFTER_MARKING //After marking	
Return value	MBWindowCheck			
Note	-			
Detailed example C#	<pre>try{ MBWindowCheck windowCheck = axMBActX1.GetWindowCheck(WindowCheckTimingTypes.WINDOW_CHECK_BEFORE_MARKING); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>Dim windowCheck As MBWindowCheck windowCheck = MBActX1.GetWindowCheck(WINDOW_CHECK_BEFORE_MARKING)</pre>			

GetXYTracking

Gets the XY tracking setting information object.

Type	Method	Mode	Operation	-
Memory allocation	✓		Editing	✓
Event	-	Blocking communication	-	
Argument	int	trackingNo	XY tracking No. (0 to 7)	
Return value	MBXYTracking	-	-	
Note	If the applicable tracking setting does not exist, “No Valid XY tracking setting exists” will occur.			
Detailed example C#	<pre>try{ MBXYTracking xyTracking = axMBActX.GetXYTracking(0); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>Dim xyTracking As MBXYTracking xyTracking = MBActX1.GetXYTracking(0)</pre>			

GetZTracking

Gets the Z tracking setting information object.

Type	Method	Mode	Operation	-
Memory allocation	✓		Editing	✓
Event	-	Blocking communication	-	
Argument	int	trackingNo	Z tracking No. (0 to 255)	
Return value	MBZTracking	-	-	
Note	If the applicable tracking setting does not exist, “No Valid Z tracking setting exists” will occur.			
Detailed example C#	<pre>try{ MBZTracking zTracking = axMBActX.GetZTracking(0); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>Dim zTracking As MBZTracking zTracking = MBActX1.GetZTracking(0)</pre>			

Property

Context

Switches between editing mode and operation mode.

Type	Property (RW)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	ContextTypes	CONTEXT_EDITING, (return value : 0)		
		CONTEXT_CONTROLLER, (return value : 1)		
Note	The return value is 0/1 in sequence from the top. CONTEXT_EDITING ... Editing mode • When the unit is offline, this property is always in this mode. • The preview displays the settings opened by OpenJobMethod or OpenControllerJobMethod. • You can edit the setting opened above by accessing the each instance of MBJob/MPBlock/MBMarkingCotion/MBMatrix/MBMatrixCell. You can use SaveJobMethod or SaveControllerJobMethod to save the edited settings. CONTEXT_CONTROLLER ... operation mode • This mode can be entered only when the unit is online. • The preview shows the current setting of the controller. • If the mark data has changed because an update character such as a counter was updated or the setting was switched, the drawing is automatically updated regardless of the value of IsAutoRedraw. • OpenJobMethod or OpenControllerJobMethod cannot be used to open a job for editing. • The finder display, if shown by "StartCameraScanningMode2", is closed. Exception: With the unit offline, when the mode is switched to the controller operation monitor mode A connection error occurs.			
Detailed example C#	axMBActX1.Context = ContextTypes.CONTEXT_EDITING;			
Simple example VBA	MBActX1.Context = CONTEXT_EDITING			

PreviewOperation

Changes the cursor operation on the preview.

Type	Property (RW)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	PreviewOperation Types	PREVIEWOPERATION_NORMAL //Basic cursor (return value : 0)		
		PREVIEWOPERATION_ZOOMUP //Zoom In lens (click to zoom in) (return value : 1)		
		PREVIEWOPERATION_ZOOMDOWN //Zoom Out lens (click to zoom out) (return value : 2)		
		PREVIEWOPERATION_NORMAL //ZOOM in lens for specified range (click to zoom in the specified range) (return value : 3)		
Return value	-			
Note	The return value is 0 to 3 in order from the top.			
Detailed example C#	axMBActX1.PreviewOperation = MBPLib1.PreviewOperationTypes.PREVIEWOPERATION_ZOOMUP;			
Simple example VBA	MBActX1.PreviewOperation = PREVIEWOPERATION_ZOOMUP MsgBox MBActX1.PreviewOperation			

Job

Gets the program setting information object.

Type	Property(R)	Mode	Operation	-
Memory allocation	✓		Editing	✓
Event	-	Blocking communication	-	
Type	MBJob	Setting information object		
Return value	-			
Note	-			
Example	See the MBOperation object.			

Operation

Gets the operation information object.

Type	Property(R)	Mode	Operation	✓
Memory allocation	✓		Editing	✓
Event	-	Blocking communication	-	
Type	MBOperation		Operation information object	
Return value	-			
Note	-			
Example	See the MBOperation object.			

Equipment

Gets the unit setup object.

Type	Property(R)	Mode	Operation	✓
Memory allocation	✓		Editing	✓
Event	-	Blocking communication	-	
Type	MBEquipment	Unit setup object		
Return value	-			
Note	-			
Example	See the MBEquipment object.			

Comm

Gets the communication management object.

Type	Property(R)	Mode	Operation	✓
Memory allocation	✓		Editing	✓
Event	-	Blocking communication	-	
Type	MBComm	Communication management object		
Return value	-			
Note	-			
Example	See the MBComm object.			

CommonBlock

Gets the common parameter object for blocks.

Type	Property(R)	Mode	Operation	✓
Memory allocation	✓		Editing	✓
Event	-	Blocking communication	-	
Type	MBCommonBlock : Common parameter for blocks object			
Return value	-			
Note	-			
Example	See the MBCommonBlock object.			

LibraryVersion

Gets the DLL version number.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	string	Character string of the version number		
Note	This number is the same as what can be obtained from MBCom.			
Detailed example C#	MessageBox.Show(axMBActX1.LibraryVersion);			
Simple example VBA	Debug.Print MBActX1.LibraryVersion			

MarkingConditionCount

Gets the number of marking conditions clipboards.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Number of marking conditions clipboards		
Note	-			
Example	See the MBMarkingCondition object.			

Dxf

Gets the DXF conversion object.

Type	Property(R)	Mode	Operation	✓
Memory allocation	✓		Editing	✓
Event	-	Blocking communication	-	
Type	MBDxf	DXF conversion setting management object		
Note	-			
Example	See the MBDxf object.			

MarkingUnit

Gets the model information.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	MarkingUnitTypes	MARKINGUNIT_MDX2000 (return value : 0)		
		MARKINGUNIT_MDX2020 (return value : 1)		
		MARKINGUNIT_MDX2500 (return value : 2)		
		MARKINGUNIT_MDX2520 (return value : 3)		
		MARKINGUNIT_MDX2050 (return value : 4)		
		MARKINGUNIT_MDX400 (return value : 5)		
		MARKINGUNIT_MDX450 (return value : 6)		
Return value	-			
Note	The return value is 0 to 6 in order from the top. The default is "MARKINGUNIT_MDX2000".			
Detailed example C#	if (axMBActX1.MarkingUnit == MBPLib1.MarkingUnitTypes.MARKINGUNIT_MDX2000){ //TODO Write the processing to be performed if the condition is true. }			
Simple example VBA	Dim A As MBPLib1.MarkingUnitTypes A = MBActX1.MarkingUnit			

IsBlockingCommunication

Gets/sets whether to communicate with the controller in the Blocking communication.

Type	Property (RW)		Mode	Operation	✓
Memory allocation	-			Editing	✓
Event	-		Blocking communication	-	
Type	bool	True	Communicate in the Blocking communication method		
		False	Communicate in the Non-Blocking communication method		
Note	If True				
	- When executing a method including expansion processing or marking processing, the method does not exit until the expansion/marketing is finished. - Although job implementation will be easier, parallel processing may be difficult.				
Note	If False				
	- When executing a method including expansion processing or marking processing, the method exits before the expansion/marketing is finished. - Parallel processing is possible, but an event handler must be securely designed. The default is “true” (Blocking communication method).				
Detailed example C#	<pre>axMBActX1.IsBlockingCommunication = true; //Set to the Blocking communication method axMBActX1.Operation.SetCurrentJobNo(2); //Switching of the current setting number. At this time, this method does not exit until the expansion is completed. axMBActX1.Operation.StartMarking(); The method is executed after the expansion processing is completed, causing no busy error.</pre>				
Simple example VBA	MBActX1.IsBlockingCommunication = True				

IsAutoRedraw

Gets/sets whether to update the drawing automatically when the property has been changed.

Type	Property (RW)		Mode	Operation	✓
Memory allocation	-			Editing	✓
Event	-		Blocking communication	-	
Type	bool	True	Automatically update		
		False	Not automatically update		
Note	The default is “true” (update the drawing automatically). In operation mode, even if IsAutoRedraw = False, the drawing is updated automatically when editing character, countering, updating character or switching settings.				
Detailed example C#	axMBActX1.IsAutoRedraw = false;				
Simple example VBA	MBActX1.IsAutoRedraw = True				

Event

EvExpansionStart

This event occurs when expansion has started.

Type	Event	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Note	After expansion has started, it may take up to 500 ms before sending EvExpansionStart.			
Detailed example C#	<pre>private void axMBActX1_EvExpansionStart(object sender, EventArgs e) { Message.Show("Expansion Start."); }</pre>			
Simple example VBA	<pre>Private Sub MBActX1_EvExpansionStart() End Sub</pre>			

EvExpansionEnd

This event occurs when expansion has been completed.

Type	Event	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Note	After expansion has started, it may take up to 500 ms before sending EvExpansionEnd.			
Detailed example C#	<pre>private void axMBActX1_EvExpansionEnd(object sender, EventArgs e) { Message.Show("Expansion Completed."); }</pre>			
Simple example VBA	<pre>Private Sub MBActX1_EvExpansionEnd() End Sub</pre>			

EvMarkingStart2

This event occurs when marking has started.

Type	Event	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Note	After expansion has started, it may take up to 500 ms before sending EvMarkingStart2.			
Detailed example C#	<pre>private void axMBActX1_EvMarkingStart2(object sender, EventArgs e) { MessageBox.Show("Marking Start."); }</pre>			
Simple example VBA	<pre>Private Sub MBActX1_EvMarkingStart2() End Sub</pre>			

EvMarkingEnd2

This event occurs when marking has been completed.

Type	Event	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	MarkResultTypes	MarkResultType	MARKRESULT_OK //When marking is successful	
			MARKRESULT_STOP //When marking cancels	
			MARKRESULT_ERROR //When marking fails	
	int	MarkTime	Marking time (ms)	
Note	Whether the marking has been normally completed and the marking time can be obtained from the argument. After the marking has been completed, it may take up to 500 ms before sending EvMarkingEnd2.			
Detailed example C#	<pre>private void axMBActX1_EvMarkingEnd2(object sender, Ax MBPLib1._DMBActXEvents_EvMarkingEndEvent e) { if (e.IsMarkingCompleted) { Message.Show("Marking Completed. MarkTime:" + e.MarkTime + "[ms]."); } else { MessageBox.Show("Marking Stop."); } }</pre>			
Simple example VBA	Private Sub MBActX1_EvMarkingEnd2(ByVal MarkResultType As MBPLib1.MarkResultTypes, ByVal MarkTime As Long) End Sub			

EvGuideLaserMarkingStart

This event occurs when guide laser marking has started.

Type	Event	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Note	After marking has started, it may take up to 500 ms before sending EvMarkingStart.			
Detailed example C#	<pre>private void axMBActX1_EvGuideLaserMarkingEnd(object sender, EventArgs e) { MessageBox.Show("Guide Laser Marking Start."); }</pre>			
Simple example VBA	<pre>Private Sub MBActX1_EvGuideLaserMarkingStart() End Sub</pre>			

EvGuideLaserMarkingEnd

Occurs when guide laser marking has been completed.

Type	Event	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	MarkResultTypes	MarkResult Type		
			MARKRESULT_STOP //When marking cancels	
			MARKRESULT_ERROR //When marking fails	
	int	MarkTime	Marking time (ms)	
Note	Whether the marking has been normally completed and the marking time can be obtained from the argument. After the marking has been completed, it may take up to 1 s before sending EvMarkingEnd.			
Detailed example C#	<pre>private void axMBActX1_EvGuideLaserMarkingEnd(object sender, Ax MBPLib1._DMBActXEvents_EvGuideLaserMarkingEndEvent e) { if (e.IsMarkingCompleted) { Message.Show("Marking Completed. MarkTime:" + e.MarkTime + "[ms]."); } else { MessageBox.Show("Guide Laser Marking Stop."); } }</pre>			
Simple example VBA	Private Sub MBActX1_EvGuideLaserMarkingEnd(ByVal MarkResultType As MBPLib1.MarkResultTypes, ByVal MarkTime As Long) End Sub			

EvConnectionError

This event occurs when the USB/Ethernet connection is physically disconnected.

Type	Event	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Note	-			
Detailed example C#	<pre>private void axMBActX1_EvConnectionError(object sender, EventArgs e) { MessageBox.Show("Occurrence of Connection Error. Offline Now."); }</pre>			
Simple example VBA	<pre>Private Sub MBActX1_EvConnectionError() End Sub</pre>			

EvError

This event occurs when the error occurrence status has changed.

Type	Event	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Note	When the number of errors in the controller has changed, this event occurs. If this event has occurred, use UpdateErrorCount and Error to obtain the latest error status.			
Detailed example C#	<pre>private void axMBActX1_EvError(object sender, EventArgs e) { try{ int ErrorCount = axMBActX1.UpdateErrorCount(); if (0 == ErrorCount){ Message.Show("Error Cleared."); } else { string List = ""; for (int index = 0; index < ErrorCount; index++){ List = List + axMBActX1.Error(index).ErrorCode + ":" + axMBActX1.Error(index).ErrorMessage + "\n"; } MessageBox.Show("Occurring Error Changed ... \n" + List); } } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); } }</pre>			
Simple example VBA	<pre>Private Sub MBActX1_EvError() End Sub</pre>			

Ev2DCodeReadingStart

This event occurs when 2D code reading has started.

Type	Event	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Note	After 2D code reading has started, it may take up to 500 ms before sending Ev2DCodeReadingStart.			
Detailed example C#	<pre>private void axMBActX1_Ev2DCodeReadingStart(object sender, EventArgs e) { Message.Show("2DCodeReading Start."); }</pre>			
Simple example VBA	<pre>Private Sub MBActX1_Ev2DCodeReadingStart() End Sub</pre>			

Ev2DCodeReadingEnd

This event occurs when 2D code reading has been completed.

Type	Event	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Note	After 2D code reading has been completed, it may take up to 500 ms before sending Ev2DCodeReadingEnd.			
Detailed example C#	<pre>private void axMBActX1_Ev2DCodeReadingEnd(object sender, AxMBPLib1._DMBActXEvents_Ev2DCodeReadingEndEvent e) { if (e.result) { Message.Show("2DCode: [" + e.content + "], grade = " + e.grade.ToString()); } else { Message.Show("2DCode Read Error"); } }</pre>			
Simple example VBA	<pre>Private Sub MBActX1_Ev2DCodeReadingEnd(ByVal Result As Boolean, ByVal Content As String, ByVal Grade As Long) End Sub</pre>			

EvMarkingConfirmationStart

Occurs when marking verification function has started.

Type	Event	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Note	After the marking verification function has been started, it may take up to 500 ms before sending EvMarkingConfirmationStart.			
Detailed example C#	<pre>private void axMBActX1_EvMarkingConfirmationStart (object sender, EventArgs e) { Message.Show("2DCodeReading Start."); }</pre>			
Simple example VBA	<pre>Private Sub MBActX1_EvMarkingConfirmationStart() End Sub</pre>			

EvMarkingConfirmationEnd

Occurs when marking verification function has been completed.

Type	Event	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	BOOL	IsConfirm	True //When the score is larger than the threshold	
			False //When the score is less than the threshold	
	int	ComfirmScore	Score	
Note	After themarking verification function has beenfinished, it may take up to 500 ms before sending EvMarkingConfirmationEnd.			
Detailed example C#	<pre>private void axMBActX1_ EvMarkingConfirmationEnd (object sender, AxMBPLib1._DMBActXEvents_ EvMarkingConfirmationEndEvent e) { if (e.IsConfirm) { Message.Show("MARKING OK, Score = " + e.ComfirmScore.ToString()); } else { Message.Show("MARKING NG! Score = " + e.ComfirmScore.ToString()); } }</pre>			
Simple example VBA	Private Sub MBActX1_EvMarkingConfirmationEnd(ByVal IsConfirm As Boolean, ByVal ConfirmScore As Long) End Sub			

EvMarkingWorkflowEnd

It will occur when the set operation before marking, marking, and operation after marking are completed.

Type	Event	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-	-	-	
Note	Performed work flow results can be acquired.			
Detailed example C#	-			
Simple example VBA	<pre>Private Sub MBActX1_EvMarkingWorkflowEnd() End Sub</pre>			

MBComm

Method

Online

This method establishes a connection to the controller.

Type	Method	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	bool	True	Successfully connected	
		False	Failed to connect	
Note	See the following for an example in the case of USB connection. For an example in the case of Ethernet connection, see IpAddress field. Exception: If online An error during connection occurs. If executing this method with non-existent controller serial No. specified (when USB is connected) The invalid controller serial No. error occurs.			
Detailed example C#	<pre>try{ //If the application is exited without disconnected, the instance may exclusively use the online connection. In this case, release the online connection. if (axMBActX1.Comm.IsOnline) { axMBActX1.Comm.Offline(); } axMBActX1.Comm.ConnectionType = MBPLib1.ConnectionTypes.CONNECTION_USB; bool is_success = axMBActX1.Comm.Online(); if (is_success) { MessageBox.Show("Online Success"); } else { MessageBox.Show("Online Failed"); } } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Comm.Online			

Offline

Disconnects the communication.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	bool	True	Successfully disconnected	
		False	Failed to disconnect	
Note	Do not call offline during communication processing. When switched to offline, the context of MBActX is initialized to CONTEXT_EDIT (Editing mode). Make sure to include the processing to disconnect communication when exiting the application. Otherwise, the application may not able to be put online at the next startup. The Offline method can enable the instance started last time to release the online status. For details on how to implement this measure, see the example of OnlineMethod.			
Detailed example C#	<pre>try{ if (!axMBActX1.Comm.IsOnline) { MessageBox.Show("Offline Now"); } else { bool is_success = axMBActX1.Comm.Offline(); if (is_success) { MessageBox.Show("Offline Success"); } else { MessageBox.Show("Offline Failed"); } } } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Comm.Offline			

SendAndReceive

Send communication command to the controller.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	string	SendMessage	Send message	
Return value	string	ReceiveMessage	Receive message	
Note	Send communication command to the controller.			
Detailed example C#	<pre>string send_command = "EX"; // Requesting error contents string receive_response; try{ receive_response = axMBActX1.Comm. SendAndReceive(send_command); MessageBox.Show(receive_response) ; } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>Dim receive_response As String receive_response = MBActX1.Comm.SendAndReceive ("WX,JobNo=1 ") Debug.Print receive_response</pre>			

Property

ConnectionType

Selects a connection type.

Type	Property (RW)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	ConnectionTypes	CONNECTION_USB //USB connection (return value : 0)		
		CONNECTION_ETHERNET //Ethernet connection (return value : 1)		
Return value	-			
Note	The return value is 0/1 in sequence from the top. Exception: When online (only when it is set) An error during connection occurs.			
Example	For USB connection, Online For an example in the case of Ethernet connection, see IpAddress field.			

IpAddress

Gets/sets the IP address. (For Ethernet connection)

Type	Property (RW)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	string	IP address		
Return value	-			
Note	This method is valid only when the “ConnectionType” is “Ethernet”. Exception: When the ConnectionType is CONNECTION_ETHERNET and it is online (only when it is set) An error, “Cannot be set during communication”, occurs. When a character string not in the IP address format is set (not in the format of xxx.xxx.xxx) (only when it is set) An error, “An invalid IP address format”, occurs.			
Detailed example C#	<pre>string markingunit_ip_address = "192.168.1.10"; try{ //If the application is exited without disconnected, the instance may exclusively use the online connection. In this case, release the online connection. if (axMBActX1.Comm.IsOnline) { axMBActX1.Comm.Offline(); } axMBActX1.Comm.ConnectionType = MBPLib1.ConnectionTypes.CONNECTION_ETHERNET; axMBActX1.Comm.IpAddress = markingunit_ip_address; bool is_success = axMBActX1.Comm.Online(); if (is_success) { MessageBox.Show("Online Success"); } else { MessageBox.Show("Online Failed"); } } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>Dim IP As String MBActX1.Comm.IpAddress = "127.0.01" IP = MBActX1.Comm.IpAddress</pre>			

IsOnline

Gets the connection status.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	bool	True	Connected	
	bool	False	Not connected	
Return value	-			
Note	Obtains whether the connection has been established. If the application is exited without the correct disconnection procedure, the application may not be put online at the next startup. The IsOnline property gets True even if the instance that started last time remains online. For further information see the example of OnlineMethod.			
Example	See Online.			

UsbControllerSerialNo

When connected via USB, gets/sets the connected controller serial No.

Type	Property (RW)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Controller serial No.		
Return value	-			

Note

When multiple controllers are connected to one PC via USB, you can use this property to put any of these controllers online. You must make a note of each controller serial No in advance.

- If 0 is specified, only the smallest USB port number of the PC is attempted to be accessed.
- If other than 0 is specified, the USB port connected to the PC is searched one by one for the specified serial No.

If the specified serial No. is not found, an exception occurs when executing OnlineMethod.
 The default is 0, so when the connection is fixed the one-to-one connection, setting is unnecessary.
 This property is valid only when "ConnectionType" is "CONNECTION_USB".
 Exception:
 When the ConnectionType is CONNECTION_USB and it is online (only when it is set)
 An error, "Cannot be set during communication", occurs.

Detailed example C#

```
int ControllerSerialNo = 12345;

try{
    //If the application is exited without disconnected, the instance may exclusively use the online connection. In this case, release the online connection.
    if (axMBActX1.Comm.IsOnline) {
        axMBActX1.Comm.Offline();
    }

    axMBActX1.Comm.ConnectionType = MBPLib1.ConnectionTypes.CONNECTION_USB;
    axMBActX1.Comm.UsbControllerSerialNo = ControllerSerialNo;
    bool is_success = axMBActX1.Comm.Online();
}
catch (System.Runtime.InteropServices.COMException error){
    MessageBox.Show(error.Message);
}
```

Simple example VBA

```
Dim A As Integer
MBActX1.Comm.UsbControllerSerialNo = 12345
A = MBActX1.Comm.UsbControllerSerialNo
```

LibraryVersion

Gets the DLL version number.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	string	Character string of the version number		
Return value	-			
Note	-			
Detailed example C#	MessageBox.Show(axMBActX1.Comm.LibraryVersion) ;			
Simple example VBA	Debug.Print MBActX1.Comm.LibraryVersion			

MBJob

Method

SetTolerance

Sets the Z height tolerance for auto-focus and external displacement sensor adjustment.

Type	Method	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	double	UpperLimit	Upper limit of tolerance (mm)	
	double	LowerLimit	Lower limit of tolerance (mm)	
	bool	IsStopMarking WithError	Operation outside the tolerance range True : Do not mark (error output) False : Mark (warning output)	
Return value	-			
Note	This property is valid only when the correction method in the height direction is “auto focus”. Exception: If out of input range The out of input range error occurs. If the correction method for workpiece position adjustment in the height direction is “Fixed” An “unsupported error” occurs.			
Detailed example C#	<pre>double upperlimit = 10; double lowerlimit = -10; bool isStopMarking = true; try{ if(axMBActX1.Job.WorkpiecePositionZCorrection == ZCorrectTypes.ZCORRECT_AUTOFOCUS){ axMBActX1.Job.SetTolerance(upperlimit, lowerlimit, isStopMarking); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Call MBActX1.Job.SetTolerance(10, -10, True)			

GetTolerance

Gets the Z height tolerance for auto-focus and external displacement sensor adjustment.

Type	Method	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	out double	UpperLimit	Upper limit of tolerance (mm)	
	out double	LowerLimit	Lower limit of tolerance (mm)	
	out bool	IsStopMarking WithError	Operation outside the tolerance range True: Do not mark (error output) False: Mark (warning output)	
Return value	-			
Note	This property is valid only when the correction method in the height direction is “auto focus”. Exception: If the correction method in the height direction is “Fixed” An “unsupported error” occurs.			
Detailed example C#	<pre>double upperlimit; double lowerlimit; bool isStopMarking; try{ if(axMBActX1.Job.WorkpiecePositionZCorrection == ZCorrectTypes.ZCORRECT_AUTOFOCUS){ axMBActX1.Job.GetTolerance(out upperlimit, out lowerlimit, out isStopMarking); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>Dim upperlimit As Double Dim lowerlimit As Double Dim isStopMarking As Boolean Call MBActX1.Job.GetTolerance(upperlimit, lowerlimit, isStopMarking) Debug.Print "UL:" & upperlimit & "mm / LL:" & lowerlimit & "mm / StopMarkingWithError:" & isStopMarking</pre>			

Property

Title

Gets a job title.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	string	Title name		
Significant figures	-			
Note	Exception: If the length of the character string exceeds the upper limit An error occurs. If an illegal character is entered An error occurs.			
Detailed example C#	<pre>try{ axMBActX1.Job.Title = "abc"; } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Job.Title = "TEST" Debug.Print MBActX1.Job.Title			

MoveCondition

Gets moving / stationary marking condition.

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	MoveCondition Types	MOVECONDITION_STATIONARY //Standstill (return value : 0)		
		MOVECONDITION_ONTHEFLY //Moving marking (return value : 1)		
Significant figures	-			
Note	The return value is 0/1 in sequence from the top.			
Detailed example C#	<pre>try{ switch (axMBActX1.Job.MoveCondition) { ; case MOVECONDITION_STATIONARY: MessageBox.Show("STATIONARY"); break; default: MessageBox.Show("NOT STATIONARY"); break; } } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Job.MoveCondition			

StationaryTriggerDelay

Gets/sets the trigger delay.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Trigger delay (s)		
Significant figures	The tenth place digit			
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>try{ axMBActX1.Job.StationaryTriggerDelay = 1.5; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Job.StationaryTriggerDelay = 9.9 Debug.Print MBActX1.Job.StationaryTriggerDelay			

IsMarkingUpperEnergyCheckEnable

Gets/sets whether to enable the marking energy upper limit threshold.

Type	Property (RW)		Mode	Operation	-
Memory allocation	-			Editing	✓
Event	-		Blocking communication	-	
Type	bool	True	Marking energy Upper limit check is valid.		
		False	Marking energy Upper limit check is invalid.		
Significant figures	The tenth place digit				
Note	Before setting “MarkingUpperEnergyLimit”, set this parameter to true. Exception: For an unsupported model The message, “This model is not supported”, occurs.				
Detailed example C#	<pre>try{ axMBActX1.Job.IsMarkingUpperEnergyCheckEnable = true; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>				
Simple example VBA	MBActX1.Job.IsMarkingUpperEnergyCheckEnable = True Debug.Print MBActX1.Job. IsMarkingUpperEnergyCheckEnable				

MarkingUpperEnergyLimit

Sets the marking energy upper limit threshold value.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Marking energy (J)		
Significant figures	The hundredth place digit			
Note	"IsMarkingUpperEnergyCheck" must be set to true. Exception: If out of input range The out of input range error occurs. For an unsupported model The message, "This model is not supported", occurs. If "IsMarkingUpperEnergyCheck" is false An error occurs.			
Detailed example C#	<pre>try{ axMBActX1.Job.MarkingUpperEnergyLimit = 99999.99; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Job.MarkingUpperEnergyLimit = 10 Debug.Print MBActX1.Job.MarkingUpperEnergyLimit			

MarkingLowerEnergyLimit

Sets the marking energy lower limit threshold value.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Marking energy (J)		
Significant figures	The hundredth place digit			
Note	"IsMarkingLowerEnergyCheck" must be set to true. Exception: If out of input range The out of input range error occurs. For an unsupported model The message, "This model is not supported", occurs. If "IsMarkingLowerEnergyCheck" is false An error occurs.			
Detailed example C#	try{ axMBActX1.Job.MarkingLowerEnergyLimit = 0.01; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			
Simple example VBA	MBActX1.Job.MarkingLowerEnergyLimit = 5 Debug.Print MBActX1.Job.MarkingLowerEnergyLimit			

IsMarkingLowerEnergyCheckEnable

Gets/sets whether to enable the marking energy lower limit threshold.

Type	Property (RW)		Mode	Operation	-
Memory allocation	-			Editing	✓
Event	-		Blocking communication	-	
Type	bool	True	Lower limit check of marking energy is valid.		
		False	Lower limit check of marking energy is invalid.		
Significant figures	-				
Note	Before setting “MarkingUpperEnergyLimit”, set this parameter to true. Exception: For an unsupported model The message, “This model is not supported”, occurs.				
Detailed example C#	<pre>try{ axMBActX1.Job.IsMarkingLowerEnergyCheckEnable = true; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>				
Simple example VBA	MBActX1.Job.IsMarkingLowerEnergyCheckEnable = True Debug.Print MBActX1.Job. IsMarkingLowerEnergyCheckEnable				

IsContinuousMarkingEnable

Gets/sets whether to perform continuous marking.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	bool	Whether to perform continuous marking		
Significant figures	-			
Note	Exception: For a grouping target job An "unsupported error" occurs. For a matrix job An "unsupported error" occurs. MoveCondition is other than MOVECONDITION_STATIONARY An "unsupported error" occurs.			
Detailed example C#	<pre>try{ axMBActX1.Job.IsContinuousMarkingEnable = true; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Job.IsContinuousMarkingEnable = True Debug.Print MBActX1.Job.IsContinuousMarkingEnable			

ContinuousMarkingRepetition

Gets/sets the repetition of continuous marking.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-	Mode	Editing	✓
Event	-		-	-
Type	int	Blocking communication	-	-
Significant figures	Repetition			
Note	"IsContinuousMarkingEnable" must be set to true. Exception: If out of input range The out of input range error occurs. If "IsContinuousMarkingEnable" is false The "unsupported error" is returned. MoveCondition is other than MOVECONDITION_STATIONARY An "unsupported error" occurs.			
Detailed example C#	<pre>try{ axMBActX1.Job.ContinuousMarkingRepetition = 100; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>MBActX1.Job.ContinuousMarkingRepetition = 2 Debug.Print MBActX1.Job.ContinuousMarkingRepetition</pre>			

ContinuousMarkingInterval

Gets/sets the interval of continuous marking.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-	Mode	Editing	✓
Event	-		-	-
Type	double	Blocking communication	-	-
Significant figures	The tenth place digit			
Note	"IsContinuousMarkingEnable" must be set to true. Exception: If out of input range The out of input range error occurs. If "IsContinuousMarkingEnable" is false The "unsupported error" is returned. MoveCondition is other than MOVECONDITION_STATIONARY An "unsupported error" occurs.			
Detailed example C#	<pre>try{ axMBActX1.Job.ContinuousMarkingInterval = 1.5; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>MBActX1.Job.ContinuousMarkingInterval = 2 Debug.Print MBActX1.Job.ContinuousMarkingInterval</pre>			

WorkpiecePositionReferenceX

Gets/sets the movement reference point (X) for workpiece position adjustment.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-	Mode	Editing	✓
Event	-		-	-
Type	double	Blocking communication	-	-
Significant figures	3rd decimal place			
Note	Exception: If out of input range The out of input range error occurs. "XYMoveTypes" is other than "XYMOVE_STATIONARY" An "unsupported error" occurs.			
Detailed example C#	<pre>try{ axMBActX1.Job.WorkpiecePositionReferenceX = -/25000; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>MBActX1.Job.WorkpiecePositionReferenceX = 10 Debug.Print MBActX1.Job.WorkpiecePositionReferenceX</pre>			

WorkpiecePositionReferenceY

Gets/sets the movement reference point (Y) for workpiece position adjustment.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-	Mode	Editing	✓
Event	-		-	-
Type	double	Blocking communication	-	-
Significant figures	3rd decimal place			
Note	Exception: If out of input range The out of input range error occurs. When the data is displayed on the finder by "StartCameraScanningMode2" An "unsupported error" occurs. MoveCondition is other than MOVECONDITION_STATIONARY An "unsupported error" occurs.			
Detailed example C#	<pre>try{ axMBActX1.Job.WorkpiecePositionReferenceY = -/25000; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>MBActX1.Job.WorkpiecePositionReferenceY = 10 Debug.Print MBActX1.Job.WorkpiecePositionReferenceY</pre>			

WorkpiecePositionOffsetX

Gets/sets the X coordinate workpiece position adjustment offset amount.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	X coordinate		
Significant figures	3rd decimal place			
Note	Exception: If out of input range The out of input range error occurs. When the data is displayed on the finder by "StartCameraScanningMode2" An "unsupported error" occurs. MoveCondition is other than MOVECONDITION_ STATIONARY An "unsupported error" occurs.			
Detailed example C#	<pre>try{ axMBActX1.Job.WorkpiecePositionOffsetX = -/25000; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Job.WorkpiecePositionOffsetX = 10 Debug.Print MBActX1.Job.WorkpiecePositionOffsetX			

WorkpiecePositionOffsetZ

Gets/sets the Z coordinate workpiece position adjustment offset amount.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Z coordinate		
Significant figures	3rd decimal place			
Note	Exception: If out of input range The out of input range error occurs. If the correction method in the height direction is other than "Fixed" An "unsupported error" occurs.			
Detailed example C#	<pre>try{ if(axMBActX1.Job.WorkpiecePositionZCorrection == ZCorrectTypes.ZCORRECT_FIXEDVALUE){ axMBActX1.Job.WorkpiecePositionOffsetZ = -5.000; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Job.WorkpiecePositionOffsetZ = 10 Debug.Print MBActX1.Job.WorkpiecePositionOffsetZ			

WorkpiecePositionOffsetY

Gets/sets the Y coordinate workpiece position adjustment offset amount.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Y coordinate		
Significant figures	3rd decimal place			
Note	Exception: If out of input range The out of input range error occurs. When the data is displayed on the finder by “StartCameraScanningMode2” An “unsupported error” occurs. MoveCondition is other than MOVECONDITION_ STATIONARY An “unsupported error” occurs.			
Detailed example C#	<pre>try{ axMBActX1.Job.WorkpiecePositionOffsetY = -/25000; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Job.WorkpiecePositionOffsetY = 10 Debug.Print MBActX1.Job.WorkpiecePositionOffsetY			

WorkpiecePositionOffsetAngle

Gets/sets the θ angle workpiece position adjustment offset amount.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	θ angle		
Significant figures	3rd decimal place			
Note	Exception: If out of input range The out of input range error occurs. When the data is displayed on the finder by "StartCameraScanningMode2" An "unsupported error" occurs. MoveCondition is other than MOVECONDITION_STATIONARY An "unsupported error" occurs.			
Detailed example C#	<pre>try{ axMBActX1.Job.WorkpiecePositionOffsetAngle = 1.0; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Job.WorkpiecePositionOffsetAngle = 10 Debug.Print MBActX1.Job.WorkpiecePositionOffsetAngle			

WorkpiecePositionZCorrection

Get/set the Z position adjustment method

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	ZCorrectTypes	ZCORRECT_FIXEDVALUE //Fix (return value : 0) ZCORRECT_AUTOFOCUS //Auto focus (return value : 1)		
Note	The return value is 0/1 in order from the top. Exception: When setting auto focus with MoveCondition being other than MOVECONDITION_STATIONARY An "unsupported error" occurs.			
Detailed example C#	<pre>try{ axMBActX1.Job.WorkpiecePositionZCorrection = ZCorrectTypes.ZCORRECT_AUTOFOCUS; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Job.WorkpiecePositionZCorrection = ZCORRECT_AUTOFOCUS Debug.Print MBActX1.Job.WorkpiecePositionZCorrection			

AutoFocusMeasureCount

Gets/sets the number of measurements with auto focus for workpiece position adjustment.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Number of measurements with auto focus		
Note	This property is valid only when the correction method in the height direction is “auto focus”. Exception: If out of input range The out of input range error occurs. If the correction method in the height direction is other than “auto focus” An “unsupported error” occurs.			
Detailed example C#	<pre>try{ if(axMBActX1.Job.WorkpiecePositionZCorrection == ZCorrectTypes.ZCORRECT_AUTOFOCUS){ axMBActX1.Job.AutoFocusMeasureCount = 3; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Job.AutoFocusMeasureCount = 5 Debug.Print MBActX1.Job.AutoFocusMeasureCount			

MBBlock

Method

SetMarkingCondition

Sets a block with the marking conditions registered in the marking conditions clipboard.

Type	Method		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	✓			Photo	×
Argument	int		MarkingConditionNo		Marking conditions clipboard No.	
Return value	-					
Note	The marking conditions must be registered in advance in the marking conditions clipboard with Marking Builder Plus. The order of update is as follows: "laser power", "scan speed", "pulse frequency", "spot variable", "repetition", and then "fill interval". Exception: If the IsEditable property is false. An "unsupported error" occurs. If the marking conditions clipboard No. is out of input range The out of input range error occurs. If the entered marking conditions clipboard No. is larger than the registered number The out of input range error occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).SetMarkingCondition(5); Assign the marking condition No. 5 to the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Call MBActX1.Block(0).SetMarkingCondition(0)					

OverprintingCode

Gets the overprinting (multi-pass) settings of a barcode/2D code.

Type	Method		Mode		Operation	-
Memory allocation	ü				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	*1	Logo	×
	Character string arc	×	2D code	✓	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Argument	int		Index		Order index	
Return value	MBOverprintingCode		-		Overprinting setting object	
Note	If overprinting setting of the specified order does not exist, null is returned. Exception: For an unsupported block An "unsupported error" occurs. * JAN, EAN, and UPC of the barcode block are not supported.					
Detailed example C#	<pre>int blockNo = 0; int index = 3 try{ if(axMBActX1.Block(blockNo).IsOverprintingCode){ axMBActX1.Block(blockNo).OverprintingCode(index).LaserPower = 50.0; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Debug.Print MBActX1.Block(1).OverprintingCode(0).Frequency					

SetDefaultLogoSize

Resets the height and width of a logo to the default.

Type	Method		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	✓
	Character string arc	×	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	✓
	Fixed point with trigger on	×			Photo	×
Argument	-					
Return value	-					
Note	For details, see the precautions for Text. Exception: For an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ axMBActX1.Block(blockNo).SetDefaultLogoSize(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Change the logo A of Job No.0 and BlockNo.0 to logo B MBActX1.Context = CONTEXT_EDITING //Change to editing mode MBActX1.OpenControllerJob 0 //Open job MBActX1.Block(0).Text = “LogoB.mlg” //Replace logo MBActX1.Block(0).SetDefaultLogoSize //Correct logo size MBActX1.SaveControllerJob 0 //Save job MBActX1.Context = CONTEXT_CONTROLLER //Change to operation mode					

Property

IsEditable

MBActX determines whether a block is editable.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	✓			Photo	*1
Argument	bool	True	An editable block exists.			
		False	No block exists, or a photo block exists.			
Return value	-					
Note	Use this property when the block set for the block No. cannot be identified in advance. Exception: If the IsEditable property is false. An "unsupported error" occurs. * A photo block cannot be edited, so false is returned.					
Detailed example C#	if(axMBAcT1.Block(2).IsEditable){ TODO Write the editing processing of the block No. 2. }					
Simple example VBA	Debug.Print MBAcT1.Block(0).IsEditable					

LaserPower

Gets/sets the laser power.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	✓			Photo	×
Type	double		Laser power			
Significant figures	The tenth place digit					
Note	At the time of setting of LaserPower, if IsLaserPowerBlockCommon (whether to use the laser power of the common parameter for blocks) is true, this value is changed to false. Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).LaserPower = 80.0; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).LaserPower = 80 Debug.Print MBActX1.Block(0).LaserPower					

ScanSpeed

Gets/sets the scan speed.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	int		Scan speed			
Significant figures	-					
Note	At the time of setting of ScanSpeed, if IsScanSpeedBlockCommon (whether to use the scan speed of the common parameter for blocks) is true, this value is changed to false. Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).ScanSpeed = 1000; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).ScanSpeed = 1000 Debug.Print MBActX1.Block(0).ScanSpeed					

Frequency

Gets/sets the pulse frequency.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	✓			Photo	×
Type	int		Pulse Frequency			
Significant figures	-					
Note	At the time of setting of Frequency, if IsFrequencyBlockCommon (whether to use the pulse frequency of the common parameter for blocks) is true, this value is changed to false. Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).Frequency = 100; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).Frequency = 100 Debug.Print MBActX1.Block(0).Frequency					

SpotVariable

Gets/sets the spot variable.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	✓			Photo	×
Type	int		Spot variable			
Significant figures	-					
Note	At the time of setting of SpotVariable, if IsSpotVariableBlockCommon (whether to use the spot variable of the common parameter for blocks) is true, this value is changed to false. Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).SpotVariable = -10; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).SpotVariable= 100 Debug.Print MBActX1.Block(0).SpotVariable</pre>					

Repetition

Gets/sets the repetition.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	int		Repetition			
Significant figures	-					
Note	At the time of setting of Repetition, if IsRepetitionBlockCommon (whether to use the repetition of the common parameter for blocks) is true, this value is changed to false. With the deep dig amount set in advance, if the repetition is set to one time, the deep dig amount is reset. In the case of a hatch log block, when both of the repetition and the fill line repetition of the block are set to one time, the deep dig amount is reset. Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).Repetition = 3; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).Repetition= 2 Debug.Print MBActX1.Block(0).Repetition					

IsMarkingEnable

Gets/sets whether marking is enabled.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	bool		Whether to enable marking			
Significant figures	-					
Note	Exception: If the IsEditable property is false. An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).IsMarkingEnable = true; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsMarkingEnable = True Debug.Print MBActX1.Block(0).IsMarkingEnable					

QualityLevel

Gets/sets the quality level.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	✓			Photo	×
Type	QualityLevelTypes		QUALITYLEVEL_TOPSPEED		//Top speed (return value : 10)	
			QUALITYLEVEL_HIGHSPEED		//High speed (return value : 11)	
			QUALITYLEVEL_STANDARD		//Standard (return value : 12)	
			QUALITYLEVEL_HIGHQUALITY		//High quality (return value : 13)	
			QUALITYLEVEL_TOPQUALITY		//Top quality (return value : 14)	
			QUALITYLEVEL_CUSTOM		//Customize (return value : 15)	
Significant figures	-					
Note	The return value is 10 to 15 in sequence from the top. Refer to the following when the quality level is specified to barcode or 2D code, which is an item in overprinting settings. Select “Set” to overprint all. Select “Get” to get the conditions of overprinting order being 0. Exception: If the quality level is set to “Customize” An “unsupported error” occurs. If the IsEditable property is false. An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).QualityLevel = MBPLib1.QualityLevelTypes.QUALITYLEVEL_STANDARD; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).QualityLevel = QUALITYLEVEL_STANDARD Debug.Print MBActX1.Block(0).QualityLevel					

ProcessingTime

Gets/sets the on time of a fixed point mark.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	×
	Fixed point	✓	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Time(ms)			
Significant figures	-					
Note	Exception: If the IsEditable property is false. An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).ProcessingTime = 5; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).ProcessingTime = 10 Debug.Print MBActX1.Block(0).ProcessingTime					

MarkingDelayTime

Gets/sets the wait time for start marking.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	×			Photo	×
Type	double		Wait time for start marking (ms)			
Significant figures	The tenth place digit					
Note	Refer to the following when the quality level is specified to barcode or 2D code, which is an item in overprinting settings. Select "Set" to overprint all. Select "Get" to get the conditions of overprinting order being 0. Exception: For an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ axMBActX1.Block(blockNo).MarkingDelayTime = 100.0; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).MarkingDelayTime = 10 Debug.Print MBActX1.Block(0).MarkingDelayTime</pre>					

Width

Gets the width of a block.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	×	2D code	✓	Hatch logo	✓
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	×			Photo	×
Type	double		Width of a block (mm)			
Note	Exception: For an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ int width = axMBAcTX1.Block(blockNo).Width; int height = axMBAcTX1.Block(blockNo).Height; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Debug.Print MBAcTX1.Block(0).Width					

Height

Gets the height of a block.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	×	2D code	✓	Hatch logo	✓
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	×			Photo	×
Type	double		Height of a block (mm)			
Note	Exception: For an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ int width = axMBActX1.Block(blockNo).Width; int height = axMBActX1.Block(blockNo).Height; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Debug.Print MBActX1.Block(0).Height					

IsLogoAspectRatioKeep

Gets/sets whether to remain the aspect ratio of logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	✓
	Character string arc	×	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	✓
	Fixed point with trigger on	×			Photo	×
Type	Bool		True: Retain aspect ratio False: Do not retain aspect ratio			
Significant figures	-					
Note	When aspect ratio is retained, if LogoWidth or LogoHeight is changed, then the other one will be changed so that the aspect ratio is retained. Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. When false is set for IsLogoAspectRatioKeep while the fill pattern of hatch logo block is contour line An exception occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).IsLogoAspectRatioKeep = True; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsLogoAspectRatioKeep = True Debug.Print MBActX1.Block(0).IsLogoAspectRatioKeep					

LogoWidth

Gets/sets the width of a logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	✓
	Character string arc	×	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	✓
	Fixed point with trigger on	×			Photo	×
Type	double		Width of a logo (mm)			
Significant figures	3rd decimal place					
Note	Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).LogoWidth = 10.250; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).LogoWidth = 30 Debug.Print MBActX1.Block(0).LogoWidth</pre>					

LogoHeight

Gets/sets the height of a logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	✓
	Character string arc	×	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	✓
	Fixed point with trigger on	×			Photo	×
Type	double		Height of a logo (mm)			
Significant figures	3rd decimal place					
Note	Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).LogoHeight = 10.250; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).LogoHeight = 30 Debug.Print MBActX1.Block(0).LogoHeight					

X

Gets/sets the X coordinate.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	*1	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	✓			Photo	×
Type	double		X coordinate			
Significant figures	3rd decimal place					
Note	* 1 For the character string arc, the setting of the central coordinate is used. Also, the central coordinate can be set outside the marking area. For details, check the input range by model. Exception: If the IsEditable property is false. An "unsupported error" occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).X = 10.250; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).X = 30 Debug.Print MBActX1.Block(0).X</pre>					

Y

Gets/sets the Y coordinate.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	*1	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	✓			Photo	×
Type	double		Y coordinate			
Significant figures	3rd decimal place					
Note	* 1 For the character string arc, the setting of the central coordinate is used. Also, the central coordinate can be set outside the marking area. For details, check the input range by model. Exception: If the IsEditable property is false. An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).Y = 10.250; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).Y = 30 Debug.Print MBActX1.Block(0).Y</pre>					

Z

Gets/sets the Z coordinate.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	✓			Photo	×
Type	double		Z coordinate			
Significant figures	3rd decimal place					
Note	Exception: If the IsEditable property is false. An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).Z = 2.50; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).Z = 10 Debug.Print MBActX1.Block(0).Z					

IsCharSizeRatioFixed

Gets whether the font size can be specified to fixed ratio.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	Bool		True: Ratio Specification False: Not ratio Specification			
Significant figures	-					
Note	For character strings which used system font, depending on the settings either True or False returns; for character strings which used TrueType font, True always returns. Exception: If the IsEditable property is false. An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Confirm the block No. 2 If (axMBActX1.Block(2).IsCharSizeRatioFixed){ MessageBox.Show(“Char size ratio is fixed.”); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); } }</pre>					
Simple example VBA	Debug.Print MBActX1.Block(0).IsCharSizeRatioFixed					

IsCharSizeProportional

Gets whether the characters are proportional or not.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	Bool		True: proportional False: monospaced			
Significant figures	-					
Note	For character strings which used system font, depending on the settings either True or False returns; for character strings which used TrueType font, True always returns. Exception: If the IsEditable property is false. An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Confirm the block No. 2 If (axMBActX1.Block(2). IsCharSizeProportional){ MessageBox.Show(“Char size is proportional.”); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Debug.Print MBActX1.Block(0).IsCharSizeProportional					

CharWidth

Gets/sets the width of a character.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Width of a character			
Significant figures	The hundredth place digit (ratio specification and proportional) The thousandth place digit (not specification)					
Note	It is the proportion to character height when IsCharSizeRatioFixed is True. The unit is %. It is character width when IsCharSizeRatioFixed is False. The unit is mm. Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 If (axMBActX1.Block(2).IsCharSizeRatioFixed){ //Specify a width of a character as a proportion to a height of a character with "%" axMBActX1.Block(2).CharWidth = 66.0; } else { //Specify a width of a character with "mm" axMBActX1.Block(2).CharWidth = 2.000; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).CharWidth = 3 Debug.Print MBActX1.Block(0).CharWidth					

CharHeight

Gets/sets the height of a character.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Height of a character			
Significant figures	3rd decimal place					
Note	Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).CharHeight = 3.000; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).CharHeight = 3 Debug.Print MBActX1.Block(0).CharHeirht					

CharSpace

Gets/sets the space between characters.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Space			
Significant figures	The hundredth place digit (ratio specification) The thousandth place digit (not ratio specification)					
Note	Set the CharAlignType to CHARALIGN_SPACE before accessing the data. If the CharAlignType is other than CHARALIGN_SPACE, an exception occurs. It is the proportion to character width when IsCharSizeRatioFixed is True. The unit is %. Also when IsCharSizeProportional is True, it becomes the proportion to the character width of alphabet capital letters. It is interval when IsCharSizeRatioFixed is False. The unit is mm. Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if(!axMBActX1.Block(2). CharAlignType == CharAlignTypes.CHARALIGN_SPACE){ If (axMBActX1.Block(2).IsCharSizeRatioFixed){ //Specify a width of a character as a proportion to a height of a character with “%” axMBActX1.Block(2).CharSpace = 16.7; } else { //Specify a width of a character with “mm” axMBActX1.Block(2).CharSpace = 0.500; } } } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).CharSpace = 4 Debug.Print MBActX1.Block(0).CharSpace					

CharPitch

Gets/sets character pitch.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Character pitch			
Significant figures	3rd decimal place					
Note	Only the TrueType block of horizontal character strings is valid. Set CharAlignType to CHARALIGN_PITCH before accessing the data. If the CharAlignType is other than CHARALIGN_PITCH, an exception occurs. Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if(!axMBActX1.Block(2). CharAlignType == CharAlignTypes.CHARALIGN_PITCH){ axMBActX1.Block(2).CharPitch = 2.500; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).CharPitch = 4 Debug.Print MBActX1.Block(0).CharPitch</pre>					

Angle

Gets/sets the block angle.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	×	2D code	✓	Hatch logo	✓
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	×			Photo	×
Type	double		Block angle			
Significant figures	3rd decimal place					
Note	Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).Angle = 30; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).Angle = 15 Debug.Print MBActX1.Block(0).Angle</pre>					

IsFillMarkingConditionEnable

Gets/sets whether to use the fill marking condition.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	Bool		True: The fill marking condition is enabled. False: The fill marking condition for a block is disabled.			
Note	This property is valid only when the character string is TrueType font. (Only valid when FontType property is FONT_TTF) Unsupported when using system font. Exception: For an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsFillMarkingConditionEnable) { //Get/set the fill marking condition. } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsFillMarkingConditionEnable = True Debug.Print MBActX1.Block(0).IsFillMarkingConditionEnable					

FillLaserPower

Gets/sets the fill line laser power of TrueType font and hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	Double		Laser power			
Significant figures	The tenth place digit					
Note	Set the IsFillMarkingConditionEnable to true before accessing the data. Accessing the data with the IsFillMarkingConditionEnable set to false causes an exception. This property is valid only when the character string is TrueType font. (Only valid when FontType property is FONT_TTF) Unsupported when using system font. Exception: If the IsEditable property is false. An “unsupported error” occurs. If the IsFillMarkingConditionEnable is false An “unsupported error” occurs. If out of input range An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if(axMBActX1.Block(2).IsFillMarkingConditionEnable){ axMBActX1.Block(2).FillLaserPower = 80.0; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).FillLaserPower = 80 Debug.Print MBActX1.Block(0).FillLaserPower</pre>					

FillScanSpeed

Gets/sets the fill line scan speed of TrueType font and hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	int		Fill line scan speed			
Significant figures	-					
Note	Set the IsFillMarkingConditionEnable to true before accessing the data. Accessing the data with the IsFillMarkingConditionEnable set to false causes an exception. This property is valid only when the character string is TrueType font. (Only valid when FontType property is FONT_TTF) Unsupported when using system font. Exception: If the IsEditable property is false. An "unsupported error" occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if(axMBActX1.Block(2).IsFillMarkingConditionEnable){ axMBActX1.Block(2).FillScanSpeed = 1000; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).FillScanSpeed = 1000 Debug.Print MBActX1.Block(0).FillScanSpeed</pre>					

FillFrequency

Gets/sets the fill pulse frequency of TrueType font and hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	int		Fill pulse frequency			
Significant figures	-					
Note	Set the IsFillMarkingConditionEnable to true before accessing the data. Accessing the data with the IsFillMarkingConditionEnable set to false causes an exception. This property is valid only when the character string is TrueType font. (Only valid when FontType property is FONT_TTF) Unsupported when using system font. Exception: If the IsEditable property is false. An "unsupported error" occurs. If the IsFillMarkingConditionEnable is false An "unsupported error" occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if(axMBActX1.Block(2).IsFillMarkingConditionEnable){ axMBActX1.Block(2).FillFrequency = 100; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).FillFrequency = 100 Debug.Print MBActX1.Block(0).FillFrequency					

FillSpotVariable

Gets/sets the fill spot variable of TrueType font and hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	bool		Fill line spot variable			
Significant figures	-					
Note	Set the IsFillMarkingConditionEnable to true before accessing the data. Accessing the data with the IsFillMarkingConditionEnable set to false causes an exception. This property is valid only when the character string is TrueType font. (Only valid when FontType property is FONT_TTF) Unsupported when using system font. Exception: If the IsEditable property is false. An “unsupported error” occurs. If the IsFillMarkingConditionEnable is false An “unsupported error” occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if(axMBActX1.Block(2).IsFillMarkingConditionEnable){ axMBActX1.Block(2).FillSpotVariable = -10; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).FillSpotVariable = 20 Debug.Print MBActX1.Block(0).FillSpotVariable					

FillRepetition

Gets/sets the fill line repetition of TrueType font and hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	✓
	Fixed point with trigger on	×			Photo	×
Type	int		Fill line repetition			
Significant figures	-					
Note	Set the IsFillMarkingConditionEnable to true before accessing the data. Accessing the data with the IsFillMarkingConditionEnable set to false causes an exception. This property is valid only when the character string is TrueType font. (Only valid when FontType property is FONT_TTF) Unsupported when using system font. Exception: If the IsEditable property is false. An “unsupported error” occurs. If the IsFillMarkingConditionEnable is false An “unsupported error” occurs. If out of input range An input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if(axMBActX1.Block(2).IsFillMarkingConditionEnable){ axMBActX1.Block(2).FillRepetition = 3; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).FillRepetition = 2 Debug.Print MBActX1.Block(0).FillRepetition</pre>					

FillType

Gets/sets the fill type of a hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	FillTypes		FILL_SLANT //Slanting line (return value : 0)			
			FILL_CROSS //Crossing (return value : 1)			
			FILL_CONTOUR //Contour line (return value : 2)			
Significant figures	-					
Note	The return value is 0 to 2 in order from the top. Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).FillType = MBPLib1.FillTypes.FILL_CONTOUR; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).FillType = FILL_CONTOUR Debug.Print MBActX1.Block(0).FillType					

FillLineInterval

Gets/sets the fill interval of a hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	✓	Logo	×
	Character string arc	×	2D code	✓	Hatch logo	✓
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	int		Fill interval(μm)			
Significant figures	-					
Note	Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If FillType is Contour An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).FillLineInterval = 40; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message);} </pre>					
Simple example VBA	MBActX1.Block(0).FillLineInterval = 50 Debug.Print MBActX1.Block(0).FillLineInterval					

FillMargin

Gets/sets the fill margin of a hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	✓	Logo	×
	Character string arc	×	2D code	✓	Hatch logo	✓
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	int		Fill margin			
Significant figures	-					
Note	Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If FillType is Contour An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).FillMargin = 0.020; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).FillMargin = 10 Debug.Print MBActX1.Block(0).FillMargin					

FillAngle

Gets/sets the fill angle of a hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	int		Fill angle			
Significant figures	-					
Note	Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If FillType is Contour An "unsupported error" occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ if (axMBActX1.Block(2).FillType != MBPLib1.FillTypes.FILL_CONTOUR) { axMBActX1.Block(2).FillAngle = 45; // Edit the block No. 2 } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).FillAngle = 15 Debug.Print MBActX1.Block(0).FillAngle</pre>					

FillCrossAngle

Gets/sets the fill cross angle of a hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	int		Fill cross angle			
Significant figures	-					
Note	Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If FillType is Contour An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if (axMBActX1.Block(2).FillType == MBPLib1.FillTypes.FILL_CROSS) { axMBActX1.Block(2).FillCrossAngle = 45; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).FillCrossAngle = 10 Debug.Print MBActX1.Block(0).FillCrossAngle					

FillDirection

Gets/sets the fill direction of a hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	int	FILLDIRECTION_LEFTWARD		//R -> L (return value : 0)		
		FILLDIRECTION_RIGHTWARD		//L -> R (return value : 1)		
		FILLDIRECTION_RIGHTWARDLEFTWARD		//L -> R, R -> L (return value : 2)		
		FILLDIRECTION_LEFTWARDRIGHTWARD		//R -> L, L -> R (return value : 3)		
		FILLDIRECTION_CLOCKWISE		//CW (return value : 4)		
		FILLDIRECTION_COUNTERCLOCKWISE		//CCW (return value : 5)		
		FILLDIRECTION_CLOCKWISECOUNTERCLOCKWISE		//CW -> CCW (return value : 6)		
		FILLDIRECTION_COUNTERCLOCKWISECLOCKWISE		//CCW -> CW (return value : 7)		
		FILLDIRECTION_RIGHTWARDLEFTWARD_FAST		//L -> R, R -> L (high speed) (return value : 8)		
		FILLDIRECTION_LEFTWARDRIGHTWARD_FAST		//R -> L, L -> R (high speed) (return value : 9)		
Significant figures	-					
Note	The return value is 0 to 9 in order from the top. Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. When FillType is FILL_CONTOUR, if a value other than “***CLOCKWISE” is set The out of input range error occurs. When FillType is FILL_SLANT, FILL_CROSS, if “***CLOCKWISE” is set The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if (axMBActX1.Block(2).FillType == MBPLib1.FillTypes.FILL_CONTOUR) { axMBActX1.Block(2).FillDirection = MBPLib1.FillDirectionTypes.FILLDIRECTION_CLOCKWISE; } else { //Edit the block No. 2 axMBActX1.Block(2).FillDirection = MBPLib1.FillDirectionTypes.FILLDIRECTION_LEFTWARD; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).FillDirection = FILLDIRECTION_CLOCKWISE Debug.Print MBActX1.Block(0).FillDirection					

FillSkipLineCount

Gets/sets the fill skip line count of a hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	int		Fill skip line count			
Significant figures	-					
Note	Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).FillSkipLineCount = 3; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).FillSkipLineCount = 2 Debug.Print MBActX1.Block(0).FillSkipLineCount</pre>					

FillStartPosition

Gets/sets the fill start position of a hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×		✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	FillStartPosTypes		FILLSTARTPOS_INNER (return value : 0)			
			FILLSTARTPOS_OUTER (return value : 1)			
Significant figures	-					
Note	The return value is 0/1 in sequence from the top. Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If FillType is other than Contour An "unsupported error" occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if (axMBActX1.Block(2).FillType == MBPLib1.FillTypes.FILL_CONTOUR) { axMBActX1.Block(2).FillStartPosition = MBPLib1.FillStartPosTypes.FILLSTARTPOS_INNER; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).FillStartPosition = FILLSTARTPOS_INNER Debug.Print MBActX1.Block(0).FillStartPosition					

IsFillOverprinting

Gets/sets the fill line overwriting ON/OFF status of the hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	bool		True		Overprinting is ON	
			False		Overprinting is OFF	
Significant figures	-					
Note	Exception: If the IsEditable property is false. An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).IsFillOverprinting = true; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsFillOverprinting = True Debug.Print MBActX1.Block(0).IsFillOverprinting					

FillOverprinting

Gets/sets the fill overprinting direction of a hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	FillOverprintingTypes		FILLOVERPRINTING_ALTERNATE //Alternate (return value : 0) FILLOVERPRINTING_FOWARDDIRECTION //Forward direction (return value : 1)			
Significant figures	-					
Note	The return value is 0/1 in sequence from the top. Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).FillOverprinting = MBPLib1.FillOverprintingTypes.FILLOVERPRINTING_ALTERNATE; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).FillOverprinting = FILLOVERPRINTING_ALTERNATE Debug.Print MBActX1.Block(0).FillOverprinting					

FillOverprintingCount

Gets/sets the fill overprinting count of a hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	int		Fill overprinting count			
Significant figures	-					
Note	Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBAcX1.Block(2).FillOverprintingCount = 3; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBAcX1.Block(0).FillOverprintingCount = 2 Debug.Print MBAcX1.Block(0).FillOverprintingCount					

FillBoundaryOffset

Gets/sets the fill boundary offset of a hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	int		Fill boundary offset			
Significant figures	-					
Note	Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBAcX1.Block(2).FillBoundaryOffset = 30; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBAcX1.Block(0).FillBoundaryOffset = 10 Debug.Print MBAcX1.Block(0).FillBoundaryOffset</pre>					

FillBoundaryOrder

Gets/sets the fill boundary order type of a hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	FillBoundaryOrderTypes		FILLBOUNDARYORDER_BOUNDARYFIRST //From boundary (return value : 0) FILLBOUNDARYORDER_FILLFIRST //From fill (return value : 1)			
Significant figures	-					
Note	The return value is 0/1 in sequence from the top. Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If IsCreateFillBoundary is false An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	try{ //Edit the block No. 2 if (axMBActX1.Block(2).IsCreateFillBoundary){ axMBActX1.Block(2).FillBoundaryOrder = MBPLib1.FillBoundaryOrderTypes.FILLBOUNDARYORDER_BOUNDARYFIRST; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }					
Simple example VBA	MBActX1.Block(0).FillBoundaryOrder = FILLBOUNDARYORDER_BOUNDARYFIRST Debug.Print MBActX1.Block(0).FillBoundaryOrder					

IsCreateFillBoundary

Gets/sets whether to create the fill boundary of a hatch logo.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	bool		True		Create a boundary	
			False		Not create a boundary	
Significant figures	-					
Note	When IsEditable is false, using other property causes the “Unsupported error”. Exception: If the IsEditable property is false. An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).IsCreateFillBoundary = true; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsCreateFillBoundary = True Debug.Print MBActX1.Block(0).IsCreateFillBoundary					

BarcodeHeight

Gets/sets the height of a barcode.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	✓	Logo	×
	Character string arc	×	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Height of a barcode			
Significant figures	3rd decimal place					
Note	Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBAcX1.Block(2).BarcodeHeight = 3.000; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBAcX1.Block(0).BarcodeHeight = 10 Debug.Print MBAcX1.Block(0).BarcodeHeight</pre>					

NarrowBarWidth

Gets/sets the narrow bar width of a barcode.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	✓	Logo	×
	Character string arc	×	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Narrow bar			
Significant figures	3rd decimal place					
Note	Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBAcX1.Block(2).NarrowBarWidth = 0.200; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBAcX1.Block(0).NarrowBarWidth = 0.2 Debug.Print MBAcX1.Block(0).NarrowBarWidth</pre>					

BarRatio

Gets/sets the bar ratio of a barcode.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	✓	Logo	×
	Character string arc	×	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Bar ratio			
Significant figures	The tenth place digit					
Note	Barcode: Only CODE39, ITF, 2of5, and NW7 are supported. Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).BarRatio = 2.5; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).BarRatio = 2.5 Debug.Print MBActX1.Block(0).BarRatio</pre>					

QuietZone

Gets/sets the quiet zone width of a barcode/2D code/ GS1Databar.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	✓	Logo	×
	Character string arc	×	2D code	✓	Hatch logo	×
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	int		Quiet zone width			
Significant figures	-					
Note	Only the block of Black/White Inversion settings are supported. Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).BarRatio = 2.5; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).QuietZone = 10 Debug.Print MBActX1.Block(0).QuietZone					

BlockType

Gets the type of a block.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	×			Photo	×
Type	BlockTypes		BLOCK_CHARACTER //Character string (return value : 0)			
			BLOCK_BARCODE //Barcode (return value : 2)			
			BLOCK_GS1DATABAR //GS1DataBar (return value : 3)			
			BLOCK_2DCODE //2D code (return value : 4)			
			BLOCK_LOGO //Logo (return value : 5)			
			BLOCK_PHOTO //Photo (return value : 6)			
			BLOCK_PROCESSING //Fixed point (return value : 8)			
Significant figures	-					
Note	The return value is 0, 2, 3, 4, 5, 6, and 8 in sequence from the top. Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if (axMBActX1.Block(2).BlockType == MBPLib1.BlockTypes.BLOCK_BARCODE){ axMBActX1.Block(2).BarcodeHeight = 3.000; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Debug.Print MBActX1.Block(0).BlockType					

DataType

Gets the detailed type of a block.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	×			Photo	×
Type	DataTypes	DATA_CHARHORIZONTAL		//String: Horizontal (return value : 0)		
		DATA_CHARVERTICAL		//String: Vertical (return value : 1)		
		DATA_CHARINARC		//Character string: Arc (counterclockwise) (return value : 2)		
		DATA_CHAROUTARC		//Character string: Arc (clockwise) (return value : 3)		
		DATA_CODE39		//Code39 (return value : 6)		
		DATA_ITF		//ITF (return value : 7)		
		DATA_2OF5		//2 of 5 (return value : 8)		
		DATA_NW7		//NW7 (return value : 9)		
		DATA_JAN		//JAN (return value : 10)		
		DATA_CODE128		//CODE128 (return value : 11)		
		DATA_QRMODEL1		//QR1 (return value : 12)		
		DATA_QRMODEL2		//QR2 (return value : 13)		
		DATA_MICROQR		//Micro QR (return value : 14)		
		DATA_DATAMATRIX		//Data matrix (return value : 15)		
		DATA_GS1DATAMATRIX		//GS1 data matrix (return value : 16)		
		DATA_GS1DATABAR		//Truncated (return value : 17)		
		DATA_GS1DATABARCCA		//Truncated-CCA (return value : 18)		
		DATA_GS1DATABARSTACKED		//Stacked (return value : 19)		
		DATA_GS1DATABARSTACKEDCCA		//Stacked-CCA (return value : 20)		
		DATA_GS1DATABARLIMITED		//Limited (return value : 21)		
		DATA_GS1DATABARLIMITEDCCA		//Limited-CCA (return value : 22)		
		DATA_LOGO		//Logo (return value : 23)		
		DATA_HATCHLOGO		//Hatch logo (return value : 24)		
		DATA_WORKIMAGE		//Workpiece image (return value : 25)		
		DATA_PROCESSINGPOINT		//Fixed point (return value : 26)		
		DATA_PROCESSINGTRIGGERONPOINT		//Fixed point with trigger on (return value : 27)		
		DATA_CODE93		//CODE93 (return value : 32)		
Significant figures	-					
Note	The return value is 0 to 3, 6 to 27, and 32 in sequence from the top. Exception: If the IsEditable property is false. An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Display the detailed type of the block No. 2; if (axMBActX1.Block(2).DataType == MBPLib1.DataTypes.DATA_CHARHORIZONTAL){ MessageBox.Show("String: Horizontal"); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Debug.Print MBActX1.Block(0).DataType					

Text

Gets/sets the character string, logo, Barcode or 2D code.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	×			Photo	×
Type	string		Character string			
Significant figures	-					
Note	The mark data can be set and obtained via editing mode. (To change in the operation mode, use SetCharString2.) When setting a logo file, specify a file name. (Eg.: "Test.mlg") When only specifying the file name, search from the logos in "C:\Users\[User name]\Documents\Marking Builder Plus\Logo" folder. To set logos other than those saved in this logo folder, specify an absolute path. Only the file name (Test.mlg), not the path name, is displayed at the time of getting of the name. Precautions) It cannot be set to a different block type, such as change the character string to a logo. If the logo is changed, the size of the original replacement will be retained. (Example : for the block in which log A has been set, if another logo B is replaced by using Text, the size of logo B is replaced after zooming in/out to the size of logo A.) To return to the specific size of the logo file, use SetDefaultLogoSize. (See the following VBA example 2) The logo size is determined based on the resolution of the selected model when saving via logo designer or during DXF conversion. Therefore, when the standard area type (resolution: 2 um) is selected, if the saved logo data is called up by the wide area type (resolution: 5 um), the size becomes 2.5 times bigger. Exception: If the IsEditable property is false. An "unsupported error" occurs. Character string is over-extended, prohibited characters are used. An error occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).Text = "abc"; //Change the character string of the block No. 2. }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); } try{ axMBActX1.Block(3).Text = "LogoFileName.MLG"; //Change the logo file of the block No. 3 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Example 1:Change the character string "ABC" of job No.0, BlockNo.0 to "123" <pre>MBActX1.Context = CONTEXT_EDITING //Change to editing mode MBActX1.OpenControllerJob 0 //Open job MBActX1.Block(0).Text = "123"//Changing character string MBActX1.SaveControllerJob 0 //Save job MBActX1.Context = CONTEXT_CONTROLLER //Change to operation mode</pre> Eg.2: Change the logo A of Job No.0 and BlockNo.0 to logo B <pre>MBActX1.Context = CONTEXT_EDITING //Change to editing mode MBActX1.OpenControllerJob 0 //Open job MBActX1.Block(0).Text = "LogoB.mlg" //Replace logo MBActX1.Block(0).SetDefaultLogoSize //Correct logo size MBActX1.SaveControllerJob 0 //Save job MBActX1.Context = CONTEXT_CONTROLLER //Change to operation mode</pre>					

IsCheckDigit

Gets/sets the check digit of a barcode.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	✓	Logo	×
	Character string arc	×	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	bool		Check digit			
Significant figures	-					
Note	Exception: If the IsEditable property is false. An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).IsCheckDigit = true; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsCheckDigit = True Debug.Print MBActX1.Block(0).IsCheckDigit					

IsLocatedOn3DShape

Gets whether a block is placed on a 3D shape.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	×			Photo	×
Type	bool		True: The block shape is 3D. False: The block shape is XY plane.			
Note	Exception: For an unsupported block An “unsupported error” occurs.					
Example	See ShapeNo.					

ShapeNo

Gets the 3D shape setting number.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	×			Photo	×
Type	int		3D shape setting number			
Note	This property is valid only when IsLocatedOn3DShape is True. Exception: For an unsupported block An "unsupported error" occurs. If IsLocatedOn3DShape is False An "unsupported error" occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsLocatedOn3DShape) { int shapeNo = axMBActX1.Block(blockNo).ShapeNo; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Debug.Print MBActX1.Block(0).ShapeNo					

QRVersion

Gets/sets the version of a QR code.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	✓	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	int		Getting of the QR version			
Significant figures	-					
Note	2D code: Only QR1, QR2, and MicroQR are supported. Exception: If the IsEditable property is false. An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Display the QR code version of the block No. 2.; MessageBox.Show(axMBActX1.Block(2).QRVersion); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).QRVersion = 2 Debug.Print MBActX1.Block(0).QRVersion					

QRErrorCorrectionLevel

Gets/sets the error correction level of a QR code.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	✓	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	QRErrCorrectLevelTypes		QRERRCORRECTLEVEL_L //7% (return value : 0)			
			QRERRCORRECTLEVEL_M //15% (return value : 1)			
			QRERRCORRECTLEVEL_Q //25% (return value : 2)			
			QRERRCORRECTLEVEL_H //30% (return value : 3)			
Significant figures	-					
Note	The return value is 0 to 3 in order from the top. 2D code: Only QR1, QR2, and MicroQR (Up to LevelQ) are supported. Exception: If the IsEditable property is false. An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).QRErrorCorrectionLevel = MBPLib1.QRErrCorrectLevelTypes.QRERRCORRECTLEVEL_L; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).QRErrorCorrectionLevel = QRERRCORRECTLEVEL_H Debug.Print MBActX1.Block(0).QRErrorCorrectionLevel					

CellSize

Gets/sets the cell size of a 2D code.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	✓	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Cell size			
Significant figures	3rd decimal place					
Note	Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ axMBActX1.Block(2).CellSize = 0.200; // Edit the block No. 2 }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).CellSize = 0.3 Debug.Print MBActX1.Block(0).CellSize					

SymbolSize

Gets/sets the symbol size of a DataMatrix or GS1 DataMatrix.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	✓	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	SymbolSizeTypes	SYMBOLSIZE_10×10 (return value : 0)		SYMBOLSIZE_64x64 (return value : 21)		
		SYMBOLSIZE_12x12 (return value : 1)		SYMBOLSIZE_72x72 (return value : 22)		
		SYMBOLSIZE_14x14 (return value : 2)		SYMBOLSIZE_80x80 (return value : 23)		
		SYMBOLSIZE_16x16 (return value : 3)		SYMBOLSIZE_88x88 (return value : 24)		
		SYMBOLSIZE_18x18 (return value : 4)		SYMBOLSIZE_96x96 (return value : 25)		
		SYMBOLSIZE_20x20 (return value : 5)		SYMBOLSIZE_104x104 (return value : 26)		
		SYMBOLSIZE_22x22 (return value : 6)		SYMBOLSIZE_120x120 (return value : 27)		
		SYMBOLSIZE_24x24 (return value : 7)		SYMBOLSIZE_132x132 (return value : 28)		
		SYMBOLSIZE_26x26 (return value : 8)		SYMBOLSIZE_144x144 (return value : 29)		
		SYMBOLSIZE_32x32 (return value : 9)		SYMBOLSIZE_8x18 (return value : 14)		
		SYMBOLSIZE_36x36 (return value : 10)		SYMBOLSIZE_8x32 (return value : 15)		
		SYMBOLSIZE_40x40 (return value : 11)		SYMBOLSIZE_12x26 (return value : 16)		
		SYMBOLSIZE_44x44 (return value : 12)		SYMBOLSIZE_12x36 (return value : 17)		
		SYMBOLSIZE_48x48 (return value : 13)		SYMBOLSIZE_16x36 (return value : 18)		
		SYMBOLSIZE_52x52 (return value : 20)		SYMBOLSIZE_16x48 (return value : 19)		
Significant figures	-					
Note	The return value is 0 to 13, 20 to 29, 14 to 19 in sequence from the top. 2D code: Only DataMatrix and GS1DataMatrix are supported. Exception: If the IsEditable property is false. An "unsupported error" occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBAcX1.Block(2).SymbolSize = MBPLib1.SymbolSizeTypes.SYMBOLSIZE_18x18; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBAcX1.Block(0).SymbolSize = SYMBOLSIZE_10x10 Debug.Print MBAcX1.Block(0).SymbolSize					

CodeForm

Gets the marking type (Normal/Black/White Inversion/Base) of a 2D/Barcode/GS1DataBar.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	✓	Logo	×
	Character string arc	×	2D code	✓	Hatch logo	×
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	CodeFormTypes		CODEFORM_NORMAL //Normal (return value : 0)			
			CODEFORM_REVERSE //Black/White Inversion (return value : 1)			
			CODEFORM_BASE //Base (return value : 2)			
Significant figures	-					
Note	The return value is 0 to 2 in order from the top. Base is supported only for 2D codes and barcodes other than LAN and GS1_DataBar. Exception: If the IsEditable property is false. An "unsupported error" occurs.					
Detailed example C#	<pre>try{ Get the type of the block No. 2.; MBPLib1.CodeFormTypes form; form = axMBActX1.Block(2).CodeForm; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Debug.Print MBActX1.Block(0).CodeForm					

Pattern2D

Gets/sets the 2D code pattern.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	✓	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	Pattern2DTypes		PATTERN2D_SYSTEM //Common (return value : 0)			
			PATTERN2D_1 //Pattern 1 (return value : 1)			
			PATTERN2D_2 //Pattern 2 (return value : 2)			
			PATTERN2D_3 //Pattern 3 (return value : 3)			
			PATTERN2D_4 //Pattern 4 (return value : 4)			
			PATTERN2D_5 //Pattern 5 (return value : 5)			
			PATTERN2D_6 //Pattern 6 (return value : 6)			
			PATTERN2D_7 //Pattern 7 (return value : 7)			
			PATTERN2D_A //Pattern A (return value : 8)			
			PATTERN2D_B //Pattern B (return value : 9)			
			PATTERN2D_C //Pattern C (return value : 10)			
			PATTERN2D_D //Pattern D (return value : 11)			
			PATTERN2D_E //Pattern E (return value : 12)			
			PATTERN2D_F //Pattern F (return value : 13)			
			PATTERN2D_ENTIRE1 //Entire pattern 1 (return value : 14)			
			PATTERN2D_ENTIRE2 //Entire pattern 2 (return value : 15)			
			PATTERN2D_ENTIRE3 //Entire pattern 3 (return value : 16)			
			PATTERN2D_ENTIRE4 //Entire pattern 4 (return value : 17)			
			PATTERN2D_INDIVIDUAL //Individual (return value : 18)			
			PATTERN2D_ENTIRE5 //Entire pattern 5 (return value : 28)			
			PATTERN2D_ENTIRE6 //Entire pattern 6 (return value : 29)			
Significant figures	-					
Note	The return value is 0 to 18, 28, 29 in sequence from the top. Exception: If the IsEditable property is false. An “unsupported error” occurs. When [Individual] is set. A setting error occurs When Pattern A to F is set for QR block The out of input range error occurs. When Pattern 1 to 7 is set for DataMatrix block The out of input range error occurs. When the Current is Pattern, if Base is attempted to be set A setting error occurs When the Current is Base, if Pattern is attempted to be set A setting error occurs					
Detailed example C#	<pre>try{ Get the type of the block No. 2.; axMBActX1.Block(2).Pattern2D =MBPLib1.Pattern2DTypes.PATTERN2D_ENTIRE1; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).Pattern2D = PATTERN2D_ENTIRE2 Debug.Print MBActX1.Block(0).Pattern2D					

PatternBarcode

Gets/sets the barcode pattern.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	✓	Logo	×
	Character string arc	×	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	PatternBarcodeTypes		PATTERNBARCODE_ALTERNATE //Alternate (return value : 0)			
			PATTERNBARCODE_UNIDIRECTIONAL //Unidirectional (return value : 1)			
			PATTERNBARCODE_BASEALTERNATE //Base: Alternate (return value : 2)			
			PATTERNBARCODE_BASEUNIDIRECTIONAL //Base: Unidirectional (return value : 3)			
			PATTERNBARCODE_BASEHORZALTERNATE //Base: Alternate (Horiz.) (return value : 4)			
			PATTERNBARCODE_BASEHORZUNIDIRECTIONAL //Base: Unidirectional (Horiz.) (return value : 5)			
Significant figures	-					
Note	The return value is 0 to 5 in order from the top. Exception: If the IsEditable property is false. An “unsupported error” occurs. When the Current is Pattern, if Base is attempted to be set A setting error occurs When the Current is Base, if Pattern is attempted to be set A setting error occurs If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBAcX1.Block(2).PatternBarcode = MBPLib1.PatternBarcodeTypes.PATTERNBARCODE_ALTERNATE; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBAcX1.Block(0).PatternBarcode = PATTERNBARCODE_ALTERNATE Debug.Print MBAcX1.Block(0).PatternBarcode					

PatternGS1DataBar

Gets/sets the pattern of GS1DataBar.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	PatternGS1DataBarTypes		PATTERNGS1DATABAR_ALTERNATE //Alternate (return value : 0) PATTERNGS1DATABAR_UNDIRECTIONAL //Unidirectional (return value : 1)			
Significant figures	-					
Note	The return value is 0/1 in sequence from the top. Exception: If the IsEditable property is false. An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).PatternGS1DataBar = MBPLib1.PatternGS1DataBarTypes.PATTERNGS1DATABAR_ALTERNATE; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).PatternGS1DataBar=PATTERNGS1DATABAR_ALTERNATE Debug.Print MBActX1.Block(0).PatternGS1DataBar					

LinearCodeHeight

Gets/sets the linear code height of GS1DataBar.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Linear code height			
Significant figures	3rd decimal place					
Note	Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).LinearCodeHeight = 6.600; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).LinearCodeHeight = 3 Debug.Print MBActX1.Block(0).LinearCodeHeight</pre>					

ModuleWidth

Gets/sets the module width of GS1DataBar.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Module width			
Significant figures	3rd decimal place					
Note	Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBAcX1.Block(2).ModuleWidth = 0.200; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBAcX1.Block(0).ModuleWidth = 1 Debug.Print MBAcX1.Block(0).ModuleWidth					

CompositeCodeModuleHeight

Gets/sets the 2D module height of GS1DataBar.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		2D module height			
Significant figures	3rd decimal place					
Note	This property can be used only for GS1 DataBar & CC. Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBAcX1.Block(2).CompositeCodeModuleHeight = 0.400; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBAcX1.Block(0).CompositeCodeModuleHeight = 0.4 Debug.Print MBAcX1.Block(0).CompositeCodeModuleHeight					

SeparatorHeight

Gets/sets the separator height of GS1DataBar.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Getting/setting of the separator height of GS1DataBar			
Significant figures	3rd decimal place					
Note	GS1DataBar: Support only for RSS14Stacked Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).SeparatorHeight = 0.200; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).SeparatorHeight = 0.2 Debug.Print MBActX1.Block(0).SeparatorHeight					

OverprintingCodeCount

Gets the overprinting count of a barcode/2D code.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	✓	Logo	×
	Character string arc	×	2D code	✓	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	int		Overprinting count			
Note	Except JAN and GS1 DataBar. Only when IsOverprintingCode is True, a value other than 0 is returned. (If False, 0 is returned.) Exception: For an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsOverprintingCode) { int count = axMBActX1.Block(blockNo).OverprintingCodeCount; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Debug.Print MBActX1.Block(0).OverprintingCodeCount					

IsOverprintingCode

Gets whether the overprinting count of a barcode/2D code is set.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	✓			Photo	×
Type	bool		True: Overprinting is enabled. False: Overprinting is disabled.			
Note	Except JAN and GS1 DataBar. Exception: For an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsOverprintingCode) { //Get/set overprinting } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Debug.Print MBActX1.Block(0).IsOverprintingCode					

ReferencePosition

Gets/sets the block reference point.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	×			Photo	×
Type	ReferencePositionTypes		REFERENCEPOSITION_BOTTOMLEFT //Bottom left (return value : 0)			
			REFERENCEPOSITION_BOTTOMRIGHT //Bottom right (return value : 1)			
			REFERENCEPOSITION_BOTTOMCENTER //Bottom center (return value : 2)			
			REFERENCEPOSITION_MIDDLELEFT //Middle left (return value : 3)			
			REFERENCEPOSITION_MIDDLERIGHT //Middle right (return value : 4)			
			REFERENCEPOSITION_MIDDLECENTER //Middle center (return value : 5)			
			REFERENCEPOSITION_TOPLEFT //Top left (return value : 6)			
			REFERENCEPOSITION_TOPRIGHT //Top right (return value : 7)			
			REFERENCEPOSITION_TOPCENTER //Top center (return value : 8)			
Significant figures	-					
Note	The return value is 0 to 8 in order from the top. Exception: If the IsEditable property is false. An “unsupported error” occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).ReferencePosition = MBPLib1.ReferencePositionTypes.REFERENCEPOSITION_CENTER; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).ReferencePosition = REFERENCEPOSITION_MIDDLELEFT Debug.Print MBActX1.Block(0).ReferencePosition					

SkipCross

Gets/sets the skip cross of characters.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Skip cross			
Significant figures	3rd decimal place					
Note	Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).SkipCross = 0.010; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).SkipCross = 0.01 Debug.Print MBActX1.Block(0).SkipCross</pre>					

ArcRadius

Gets/sets the arc radius of a character string arc.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Arc radius			
Significant figures	3rd decimal place					
Note	Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).ArcRadius = 20.000; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).ArcRadius = 20 Debug.Print MBActX1.Block(0).ArcRadius</pre>					

StartAngle

Gets/sets the start angle of a character string arc.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Start angle			
Significant figures	3rd decimal place					
Note	Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).StartAngle = 210.00; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).StartAngle = 90 Debug.Print MBActX1.Block(0).StartAngle					

CharSpaceAngle

Gets/sets the space angle of a character string arc.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Space angle			
Significant figures	3rd decimal place					
Note	Usable only when CharAlignType is CHARALIGN_ANGLESPACE (Angular interval). Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).CharSpaceAngle = 10.00; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).CharSpaceAngle = 10 Debug.Print MBActX1.Block(0).CharSpaceAngle					

OpenAngle

Gets/sets the open angle of a character string arc.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	×	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Open angle			
Significant figures	3rd decimal place					
Note	Usable only when CharAlignType is CHARALIGN_EQUALLAYOUT (Distribute). Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).OpenAngle = 10.00; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).OpenAngle = 45 Debug.Print MBActX1.Block(0).OpenAngle					

FontType

Gets font type of character strings.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	FontTypes		FONT_SYSTEM //System font (return value : 0)			
			FONT_TRUETYPE //TrueType font (return value : 1)			
Significant figures	-					
Note	The return value is 0/1 in sequence from the top. Exception: If the IsEditable property is false. An “unsupported error” occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Display the font type of the block No. 2; if (axMBActX1.Block(2).FontType == MBPLib1.FontTypes.FONT_TTF){ MessageBox.Show(“TrueType Font”); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Debug.Print MBActX1.Block(0).FontType					

FontNo

Gets/sets the font number of a character string.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	Int		Font number			
Significant figures	-					
Note	This property is valid only when the character string is system font. (Only valid when FontType property is FONT_SYSTEM) Unsupported when using TrueType font. Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If no font is assigned The "Invalid font number" error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).FontNo = 1; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).FontNo = 0 Debug.Print MBActX1.Block(0).FontNo</pre>					

FontName

Gets the font name of a character string.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	string		Font name			
Significant figures	-					
Note	For a null character string, no font is assigned. Exception: If the IsEditable property is false. An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Display the font name of the block No. 2 MessageBox.Show(axMBActX1.Block(2).FontName); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Debug.Print MBActX1.Block(0).FontName					

IsFontStyleBold

Gets whether the TrueType font is bold.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	Bool		True : Bold False : Not bold			
Significant figures	-					
Note	This property is valid only when the character string is TrueType font. (Only valid when FontType property is FONT_TTF) Unsupported when using system font. Exception: If the IsEditable property is false. An “unsupported error” occurs. If this property is gotten from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Display when the TrueType font of the block No. 2 is bold If (axMBActX1.Block(2).IsFontStyleBold) { MessageBox.Show(“Font Style is Bold.”); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Debug.Print MBActX1.Block(0).IsFontStyleBold					

IsFontStyleItalic

Gets whether the TrueType font is italic.

Type	Property(R)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	Bool		True : Italic False : Not italic			
Significant figures	-					
Note	This property is valid only when the character string is TrueType font. (Only valid when FontType property is FONT_TTF) Unsupported when using system font. Exception: If the IsEditable property is false. An “unsupported error” occurs. If this property is gotten from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Display when the TrueType font of the block No. 2 is bold If (axMBActX1.Block(2).IsFontStyleItalic) { MessageBox.Show(“Font Style is Italic.”); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	Debug.Print MBActX1.Block(0).IsFontStyleItalic					

CharLineType

Gets/sets the line type of a character string.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	CharLineTypes		CHARLINE_SINGLE //Single line (return value : 0)			
			CHARLINE_MULTIPLE //Bold line (return value : 1)			
			CHARLINE_WOBBLE //Wobble (return value : 2)			
Significant figures	-					
Note	The return value is 0 to 2 in order from the top. If the line type is changed from Single to Multiple, the default line width and lines are set. See Marking Builder Plus. This property is valid only when the character string is system font. (Only valid when FontType property is FONT_SYSTEM) Unsupported when using TrueType font. Exception: If the IsEditable property is false. An "unsupported error" occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).CharLineType = MBPLib1.CharLineTypes.CHARLINE_SINGLE; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).CharLineType = CHARLINE_SINGLE Debug.Print MBActX1.Block(0).CharLineType</pre>					

MultipleLineWidth

Gets/sets the multiple line width of a character string.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Multiple line width			
Significant figures	3rd decimal place					
Note	This property is valid only when CharLineType is CHARLINE_MULTIPLE (the line type is Multiple). Exception: If the IsEditable property is false. An “unsupported error” occurs. If CharLineType is other than CHARLINE_MULTIPLE An “unsupported error” occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 axMBActX1.Block(2).MultipleLineWidth = 0.200; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).MultipleLineWidth = 0.4 Debug.Print MBActX1.Block(0).MultipleLineWidth</pre>					

IsAutoMultipleLines

Gets/sets whether to set the number of multiple lines of a character string to Auto.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	bool		True: Set the number of lines to Auto. False: Do not set the number of lines to Auto.			
Significant figures	3rd decimal place					
Note	This property is valid only when CharLineType is CHARLINE_MULTIPLE (the line type is Multiple). If the Auto setting is OFF, the default of lines is set. See Marking Builder Plus. Exception: If the IsEditable property is false. An "unsupported error" occurs. If CharLineType is other than CHARLINE_MULTIPLE An error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if (axMBActX1.Block(2).CharLineType == MBPLib1.CharLineTypes.CHARLINE_MULTIPLE){ axMBActX1.Block(2).IsAutoMultipleLines = true; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsAutoMultipleLines = True Debug.Print MBActX1.Block(0).IsAutoMultipleLines					

MultipleLines

Gets/sets the number of multiple marking lines of a character string.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Number of multiple marking lines			
Significant figures	-					
Note	This property is valid only when CharLineType is CHARLINE_MULTIPLE (the line type is Multiple). Exception: If the IsEditable property is false. An “unsupported error” occurs. If CharLineType is other than CHARLINE_MULTIPLE An error occurs. If the number of multiple lines is set to Auto An error occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if (axMBActX1.Block(2).CharLineType == MBPLib1.CharLineTypes.CHARLINE_MULTIPLE){ if (!axMBActX1.Block(2).IsAutoMultipleLines){ axMBActX1.Block(2).MultipleLines = 5; } } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).MultipleLines = 4 Debug.Print MBActX1.Block(0).MultipleLines					

WobbleLineWidth

Gets/sets the Wobble line width of a character string.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Multiple line width			
Significant figures	3rd decimal place					
Note	This property is valid only when CharLineType is CHARLINE_WOBBLE (the line type is Wobble). Exception: If the IsEditable property is false. An "unsupported error" occurs. If CharLineType is other than CHARLINE_WOBBLE An error occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if (axMBActX1.Block(2).CharLineType == MBPLib1.CharLineTypes.CHARLINE_WOBBLE){ axMBActX1.Block(2).WobbleLineWidth = 0.200; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).WobbleLineWidth = 0.2 Debug.Print MBActX1.Block(0).WobbleLineWidth</pre>					

WobbleOverlapRatio

Gets/sets the Wobble overlap rate of a character string.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Multiple line width			
Significant figures	The tenth place digit					
Note	This property is valid only when CharLineType is CHARLINE_WOBBLE (the line type is Wobble). The unit is %. Exception: If the IsEditable property is false. An "unsupported error" occurs. If CharLineType is other than CHARLINE_WOBBLE An error occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if (axMBActX1.Block(2).CharLineType == MBPLib1.CharLineTypes.CHARLINE_WOBBLE){ axMBActX1.Block(2).WobbleOverlapRatio = 80.0; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(5).WobbleOverlapRatio = 80 Debug.Print MBActX1.Block(5).WobbleOverlapRatio</pre>					

CharFullWidth

Gets/sets the full width of a character string.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	×	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Full width			
Significant figures	3rd decimal place					
Note	Set CharAlignType to CHARALIGN_EQUALLAYOUT (Distribute) before accessing the data. Exception: If the IsEditable property is false. An "unsupported error" occurs. When CharAlignType is other than CHARALIGN_EQUALLAYOUT (Distribute) An error occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>try{ //Edit the block No. 2 if (axMBActX1.Block(2).CharAlignType == MBPLib1.CharAlignTypes.CHARALIGN_EQUALLAYOUT) { axMBActX1.Block(2).CharFullWidth = 20.000; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	<pre>MBActX1.Block(0).CharFullWidth = 20.000 Debug.Print MBActX1.Block(0).CharFullWidth</pre>					

CharAngle

Gets/sets the character angle.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	double		Character angle			
Significant figures	3rd decimal place					
Note	Exception: If the IsEditable property is false. An "unsupported error" occurs. If out of input range The out of input range error occurs. If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	try{ //Edit the block No. 2 axMBActX1.Block(2).CharAngle = 360.00; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }					
Simple example VBA	MBActX1.Block(0).CharAngle = 90 Debug.Print MBActX1.Block(0).CharAngle					

CharAlignType

Gets/sets the character layout (space/angular interval/distribution).

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	CharAlignTypes		CHARALIGN_SPACE		//Character space (return value : 0)	
			CHARALIGN_ANGLESPACE		//Angle space (return value : 1)	
			CHARALIGN_EQUALLAYOUT		//Distribute (in the case of character string arc: distribution angle) (return value : 2)	
			CHARALIGN_PITCH		//Character pitch (return value : 3)	
Note	The return value is 0 to 3 in order from the top. Only valid when angular interval specification is character string arc block. Only valid when the character pitch is the block which uses TrueType of horizontal character strings. Exception: If this property is set from an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ axMBActX1.Block(blockNo).CharAlignType = CharAlignTypes.CHARALIGN_EQUALITY; axMBActX1.Block(blockNo).CharFullWidth = 20; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).CharAlignType = CHARALIGN_PITCH Debug.Print MBActX1.Block(0).CharAlignType					

IsLaserPowerBlockCommon

Gets/sets whether to use the common laser power parameter for blocks.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	bool		True: The common laser power parameter is enabled. False: The laser power parameter is set by the individual block.			
Note	If IsOverprintingCode is True, this parameter is ignored while the parameter of the overprinting settings (OverprintingCode) is enabled. Exception: For an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsFillLineIntervalBlockCommon) { axMBActX1.CommonBlock.FillLineInterval = 50; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsLaserPowerBlockCommon = True Debug.Print MBActX1.Block(0).IsLaserPowerBlockCommon					

IsScanSpeedBlockCommon

Gets/sets whether to use the common scan speed parameter for blocks.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	bool		True: The common scan speed parameter is enabled. False: The scan speed parameter is set by the individual block.			
Note	If IsOverprintingCode is True, this parameter is ignored while the parameter of the overprinting settings (OverprintingCode) is enabled. Exception: For an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsScanSpeedBlockCommon) { axMBActX1.CommonBlock.ScanSpeed = 1230; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsScanSpeedBlockCommon = True Debug.Print MBActX1.Block(0).IsScanSpeedBlockCommon					

IsFrequencyBlockCommon

Gets/sets whether to use the common pulse frequency parameter for blocks.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	✓			Photo	×
Type	bool		True: The common pulse frequency parameter is enabled. False: The pulse frequency parameter is set by the individual block.			
Note	If IsOverprintingCode is True, this parameter is ignored while the parameter of the overprinting settings (OverprintingCode) is enabled. Exception: For an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsFrequencyBlockCommon) { axMBActX1.CommonBlock.Frequency = 90; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsFrequencyBlockCommon = True Debug.Print MBActX1.Block(0).IsFrequencyBlockCommon					

IsSpotVariableBlockCommon

Gets/sets whether to use the common spot variable parameter for blocks.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	✓			Photo	×
Type	bool		True: The common spot variable parameter is enabled. False: The spot variable parameter is set by the individual block.			
Note	If IsOverprintingCode is True, this parameter is ignored while the parameter of the overprinting settings (OverprintingCode) is enabled. Exception: For an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsSpotVariableBlockCommon) { axMBActX1.CommonBlock.SpotVariable = -10; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsSpotVariableBlockCommon = True Debug.Print MBActX1.Block(0).IsSpotVariableBlockCommon					

IsQualityLevelBlockCommon

Gets/sets whether to use the common quality level parameter for blocks.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	✓			Photo	×
Type	bool		True: The common quality level parameter is enabled. False: The quality level parameter is set by the individual block.			
Note	Refer to the following when the quality level is specified to barcode or 2D code, which is an item in overprinting settings. Select “Write” to overprint all. Select “Read” to get the conditions of overprinting order being 0. For the block which has been set to “Customize”, if “Get” is selected, False will appear instead of an error. Exception: For an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsQualityLevelBlockCommon) { axMBActX1.CommonBlock.QualityLevel = QualityLevelTypes.QUALITYLEVEL_STANDARD; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsQualityLevelBlockCommon = True Debug.Print MBActX1.Block(0).IsQualityLevelBlockCommon					

IsRepetitionBlockCommon

Gets/sets whether to use the common repetition parameter for blocks.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	✓			Photo	×
Type	bool		True: The common repetition parameter is enabled. False: The repetition parameter is set by the individual block.			
Note	If IsOverprintingCode is True, this parameter is ignored while the parameter of the overprinting settings (OverprintingCode) is enabled. Exception: For an unsupported block An "unsupported error" occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsRepetitionBlockCommon) { axMBActX1.CommonBlock.Repetition = 2; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsRepetitionBlockCommon = True Debug.Print MBActX1.Block(0).IsRepetitionBlockCommon					

IsZBlockCommon

Gets/sets whether to use the common Z coordinate parameter for blocks.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	✓
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	✓	GS1DataBar	✓	Workpiece image logo	✓
	Fixed point with trigger on	✓			Photo	×
Type	bool		True: The common Z coordinate parameter is enabled. False: The Z coordinate parameter is set by the individual block.			
Note	Exception: For an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsZBlockCommon) { axMBActX1.CommonBlock.Z = -1.000; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsZBlockCommon = True Debug.Print MBActX1.Block(0).IsZBlockCommon					

IsSkipCrossBlockCommon

Gets/sets whether to use the common skip cross parameter for blocks.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	×
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	bool		True: The common skip cross parameter is enabled. False: The skip cross parameter is set by the individual block.			
Note	This property is valid only when the character string is system font. (Only valid when FontType property is FONT_SYSTEM) Unsupported when using TrueType font. Exception: For an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsSkipCrossBlockCommon) { axMBActX1.CommonBlock.SkipCross = 0.500; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsSkipCrossBlockCommon = True Debug.Print MBActX1.Block(0).IsSkipCrossBlockCommon					

IsFillLaserPowerBlockCommon

Gets/sets whether to use the common fill line laser power parameter for blocks.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	bool		True: The common fill line laser power parameter is enabled. False: The fill line laser power parameter is set by the individual block.			
Note	This property is enabled only when the fill marking condition (IsFillMarkingConditionEnable) is valid. This property is valid only when the character string is TrueType font. (Only valid when FontType property is FONT_TTF) Unsupported when using system font. Exception: For an unsupported block An “unsupported error” occurs. If IsFillMarkingConditionEnable is False An “unsupported error” occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsFillMarkingConditionEnable && !axMBActX1.Block(blockNo).IsFillLaserPowerBlockCommon) { axMBActX1.Block(blockNo).FillLaserPower = 90.0; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsFillLaserPowerBlockCommon = True Debug.Print MBActX1.Block(0).IsFillLaserPowerBlockCommon					

IsFillScanSpeedBlockCommon

Gets/sets whether to use the common fill line scan speed parameter for blocks.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	bool		True: The common fill line scan speed parameter is enabled. False: The fill line scan speed parameter is set by the individual block.			
Note	This property is enabled only when the fill marking condition (IsFillMarkingConditionEnable) is valid. This property is valid only when the character string is TrueType font. (Only valid when FontType property is FONT_TTF) Unsupported when using system font. Exception: For an unsupported block An "unsupported error" occurs. If IsFillMarkingConditionEnable is False An "unsupported error" occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsFillMarkingConditionEnable && !axMBActX1.Block(blockNo).IsFillScanSpeedBlockCommon) { axMBActX1.Block(blockNo).FillScanSpeed = 1000; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsFillScanSpeedBlockCommon = True Debug.Print MBActX1.Block(0).IsFillScanSpeedBlockCommon					

IsFillFrequencyBlockCommon

Gets/sets whether to use the common fill line pulse frequency parameter for blocks.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	bool		True: The common fill line pulse frequency parameter is enabled. False: The fill line pulse frequency parameter is set by the individual block.			
Note	This property is enabled only when the fill marking condition (IsFillMarkingConditionEnable) is valid. This property is valid only when the character string is TrueType font. (Only valid when FontType property is FONT_TTF) Unsupported when using system font. Exception: For an unsupported block An "unsupported error" occurs. If IsFillMarkingConditionEnable is False An "unsupported error" occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsFillMarkingConditionEnable && !axMBActX1.Block(blockNo).IsFillFrequencyBlockCommon) { axMBActX1.Block(blockNo).FillFrequency = 110; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsFillFrequencyBlockCommon = True Debug.Print MBActX1.Block(0).IsFillFrequencyBlockCommon					

IsFillSpotVariableBlockCommon

Gets/sets whether to use the common fill line spot variable parameter for blocks.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	bool		True: The common fill line spot variable parameter is enabled. False: The fill line spot variable parameter is set by the individual block.			
Note	This property is enabled only when the fill marking condition (IsFillMarkingConditionEnable) is valid. This property is valid only when the character string is TrueType font. (Only valid when FontType property is FONT_TTF) Unsupported when using system font. Exception: For an unsupported block An "unsupported error" occurs. If IsFillMarkingConditionEnable is False An "unsupported error" occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsFillMarkingConditionEnable && !axMBActX1.Block(blockNo).IsFillSpotVariableBlockCommon) { axMBActX1.Block(blockNo).FillSpotVariable = -5; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsFillSpotVariableBlockCommon = True Debug.Print MBActX1.Block(0).IsFillSpotVariableBlockCommon					

IsFillRepetitionBlockCommon

Gets/sets whether to use the common fill line repetition parameter for blocks.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	×	Logo	×
	Character string arc	✓	2D code	×	Hatch logo	✓
	Fixed point	×	GS1DataBar	×	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	bool		True: The common fill line repetition parameter is enabled. False: The fill line repetition parameter is set by the individual block.			
Note	This property is enabled only when the fill marking condition (IsFillMarkingConditionEnable) is valid. This property is valid only when the character string is TrueType font. (Only valid when FontType property is FONT_TTF) Unsupported when using system font. Exception: For an unsupported block An "unsupported error" occurs. If IsFillMarkingConditionEnable is False An "unsupported error" occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsFillMarkingConditionEnable && !axMBActX1.Block(blockNo).IsFillRepetitionBlockCommon) { axMBActX1.Block(blockNo).FillRepetition = 2; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsFillRepetitionBlockCommon = True Debug.Print MBActX1.Block(0).IsFillRepetitionBlockCommon					

IsFillLineIntervalBlockCommon

Gets/sets whether to use the common line interval parameter for blocks.

Type	Property (RW)		Mode		Operation	-
Memory allocation	-				Editing	✓
Event	-		Blocking communication		-	
Supported block	Horizontal/Vertical	✓	Barcode	✓	Logo	×
	Character string arc	✓	2D code	✓	Hatch logo	✓
	Fixed point	×	GS1DataBar	✓	Workpiece image logo	×
	Fixed point with trigger on	×			Photo	×
Type	bool		True: The common fill interval parameter is enabled. False: The fill interval parameter is set by the individual block.			
Note	This property is valid only when the character string is TrueType font. (Only valid when FontType property is FONT_TTF) Unsupported when using system font. If IsOverprintingCode is True, this parameter is ignored while the parameter of the overprinting settings (OverprintingCode) is enabled. Exception: For an unsupported block An “unsupported error” occurs.					
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsFillLineIntervalBlockCommon) { axMBActX1.CommonBlock.FillLineInterval = 50; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					
Simple example VBA	MBActX1.Block(0).IsFillLineIntervalBlockCommon = True Debug.Print MBActX1.Block(0).IsFillLineIntervalBlockCommon					

MBMarkingCondition

Property

Caption

Gets the comments on the marking conditions clipboard.

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	string	Name		
Significant figures	-			
Note	-			
Detailed example C#	<pre>int count; string list = ""; try{ count = axMBActX1.MarkingConditionCount; for (int index = 0; index < count; index++){ list = list + "["+ index+ "]" + axMBActX1.MarkingCondition(index).Caption + "\n" + "Pw:" + axMBActX1.MarkingCondition(index).LaserPower + "% "+ "Ss:" + axMBActX1.MarkingCondition(index).ScanSpeed + "mm/s " + "Sp:" + axMBActX1.MarkingCondition(index).SpotVariable + " " + "Fr:" + axMBActX1.MarkingCondition(index).Frequency + "kHz " + "Rp:" + axMBActX1.MarkingCondition(index).Repetition + "\n"; } MessageBox.Show("Marking Condition List is below ...\n" + list); } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>Debug.Print MBActX1.MarkingCondition(0).Caption Debug.Print MBActX1.MarkingCondition(0).LaserPower Debug.Print MBActX1.MarkingCondition(0).ScanSpeed Debug.Print MBActX1.MarkingCondition(0).Frequency Debug.Print MBActX1.MarkingCondition(0).SpotVariable Debug.Print MBActX1.MarkingCondition(0).Repetition Debug.Print MBActX1.MarkingCondition(0).FillLineInterval</pre>			

LaserPower

Gets laser power from the marking conditions clipboard.

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Laser power		
Significant figures	-			
Note	-			
Example	See Caption.			

Repetition

Gets repetition from the marking conditions clipboard.

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Repetition		
Significant figures	-			
Note	-			
Example	See Caption.			

ScanSpeed

Gets scan speed from the marking conditions clipboard.

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Scan speed		
Significant figures	-			
Note	-			
Example	See Caption.			

FillLineInterval

Gets the fill line interval of the marking conditions clipboard. (unit: μm).

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Setting/getting of the fill interval		
Significant figures	-			
Note	-			
Example	See Caption.			

Frequency

Gets frequency from the marking conditions clipboard.

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Pulse Frequency		
Significant figures	-			
Note	-			
Example	See Caption.			

SpotVariable

Gets spot variable from the marking conditions clipboard.

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Spot variable		
Significant figures	-			
Note	-			
Example	See Caption.			

MBEquipment

Method

AdjustTime

Sets the internal clock to a specified date and time.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	DateTime	NewDateTime	Date and time to synchronize	
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>try{ DateTime dt = DateTime.Parse("2012/4/1 12:34:56"); axMBActX1.Equipment.AdjustTime(dt); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>Dim dt As Date dt = "2015/01/01 0:0:0" MBActX1.Equipment.AdjustTime dt</pre>			

GetControllerTime

Gets the date and time of the controller.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	DateTime	-	Date and time of the controller	
Note	Exception: If the equipment is not connected with the unit The “Communication error” occurs.			
Detailed example C#	<pre>DateTime ControllerDateTime; try{ ControllerDateTime = axMBActX1.Equipment. GetControllerTime(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Equipment.GetControllerTime			

SynchronizePcTime

Sets the date and time of a laser marker to the date and time of the connected PC.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>try{ axMBActX1.Equipment.SynchronizePcTime(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Equipment.SynchronizePcTime Debug.Print MBActX1.Equipment.GetControllerTime			

StartWarmUp

Forcibly starts warm up.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvError	Blocking communication	-	
Argument	int	WarmupTime	Warm up time (min.) [1 - 120]	
	double	WarmupInterval	Warm up interval (sec.) [0.0 - 9.9] (The tenth place digit)	
Return value	-			
Note	The terminal block is in the “Warming up” status during the execution, causing the EvError event to occur when starting and exiting. Exception: If the equipment is not connected with the unit The “Communication error” occurs. If out of input range The out of input range error occurs.			
Detailed example C#	try{ axMBActX1.Equipment.StartWarmUp(30, 1.5); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			
Simple example VBA	MBActX1.Equipment.StartWarmUp 30, 1			

StopWarmUp

Cancels the warm up.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvError	Blocking communication	-	
Argument	-			
Return value	-			
Note	The terminal block is in the “Warming up” status during the execution, causing the EvError event to occur when the warm up is cancelled. Exception: If the equipment is not connected with the unit The “Communication error” occurs.			
Detailed example C#	<pre>try{ axMBActX1.Equipment.StartWarmUp(30, 1.5); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Equipment.StopWarmUp			

GetOffsetPower

Gets the power offset value from unit setup.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	double	Power offset (The tenth place digit)		
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>double OffsetPower; try{ OffsetPower = axMBActX1.Equipment.GetOffsetPower(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Equipment.SetOffsetPower 10 Debug.Print MBActX1.Equipment.GetOffsetPower			

SetOffsetPower

Sets the power offset value in unit setup.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvExpansionStart/ End	Blocking communication	✓	
Argument	double	OffsetPower	Power offset (The tenth place digit)	
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. If out of input range The out of input range error occurs.			
Detailed example C#	<pre>try{ axMBActX1.Equipment.SetOffsetPower(2.5); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Equipment.SetOffsetPower 10 Debug.Print MBActX1.Equipment.GetOffsetPower			

SetInstallationPositionCorrection

Sets the focal distance and position correction.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvExpansionStart/End	Blocking communication		✓
Argument	double	RotationX //Rotation angle X (The thousandth place digit)		
		RotationY //Rotation angle Y (The thousandth place digit)		
		RotationZ //Rotation angle Z (The thousandth place digit)		
		CoordinateX //Shift X (The thousandth place digit)		
		CoordinateY //Shift Y (The thousandth place digit)		
		CoordinateZ //Shift Z (The thousandth place digit)		
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. If out of input range The out of input range error occurs. When the data is displayed on the finder by "StartCameraScanningMode2" An "unsupported error" occurs.			
Detailed example C#	<pre>double RotationX = 1.00; double RotationY = 2.00; double RotationZ = 3.00; double CoordinateX = 4.000; double CoordinateY = 5.000; double CoordinateZ = 6.000; try{ axMBActX1.Equipment.SetInstallationPositionCorrection(RotationX, RotationY, RotationZ, CoordinateX, CoordinateY, CoordinateZ); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>Dim RotationX As Double Dim RotationY As Double Dim RotationZ As Double Dim CoordinateX As Double Dim CoordinateY As Double Dim CoordinateZ As Double MBActX1.Equipment.SetInstallationPositionCorrection 0, 0, 0, -10, 0, 0 MBActX1.Equipment.GetInstallationPositionCorrection RotationX, RotationY, RotationZ, CoordinateX, CoordinateY, CoordinateZ Debug.Print RotationX, RotationY, RotationZ, CoordinateX, CoordinateY, CoordinateZ</pre>			

GetInstallationPositionCorrection

Gets the focal distance and position correction.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication		-
Argument	out double	RotationX //Rotation angle X (The thousandth place digit)		
		RotationY //Rotation angle Y (The thousandth place digit)		
		RotationZ //Rotation angle Z (The thousandth place digit)		
		CoordinateX //Shift X (The thousandth place digit)		
		CoordinateY //Shift Y (The thousandth place digit)		
		CoordinateZ //Shift Z (The thousandth place digit)		
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>double RotationX; double RotationY; double RotationZ; double CoordinateX; double CoordinateY; double CoordinateZ; try{ axMBActX1.Equipment.GetInstallationPositionCorrection(out RotationX, out RotationY, out RotationZ, out CoordinateX, out CoordinateY, out CoordinateZ); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	See SetInstallationPositionCorrection.			

StartPowerMonitorZeroPointCorrection

Performs the power monitor zero point correction.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication		*1
Argument	-			
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. * This method, once executed, does not exit for 5 seconds regardless of the value of IsBlockingCommunication. After 5 seconds, the method exits.			
Detailed example C#	<pre>try{ axMBActX1.Equipment.StartPowerMonitorZero PointCorrection(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>MBActX1.Equipment. StartPowerMonitorZeroPointCorrectionff</pre>			

GetLaserPowerCalibrationMode

Gets laser power calibration mode.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	out LaserCalibrationTypes	CalibrationType (correction type)	LASERCALIBRATION_DEFAULTS //Defaults	
			LASERCALIBRATION_AUTO //Auto	
			LASERCALIBRATION_MANUAL //Manual	
	out DateTime	AutoExecutionDate	Last execution date when set to Auto	
	out int	ManualCorrectionValue	Manual correction value	
Note	About DateTime of the last execution date when set to Auto Only the date is valid. The time to be obtained is fixed to 00:00:00. Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>MBPLib1.LaserCalibrationTypes mode; DateTime dt; int correction; string mode_string = ""; try{ axMBActX1.Equipment.GetLaserPowerCalibrationMode(out mode, out dt, out correction); switch(mode) { case MBPLib1.LaserCalibrationTypes.LASERCALIBRATION_DEFAULTS:mode_string = "Defaults"; break; case MBPLib1.LaserCalibrationTypes.LASERCALIBRATION_AUTO: mode_string = "Auto : executed at " + dt.ToShortDateString(); break; case MBPLib1.LaserCalibrationTypes.LASERCALIBRATION_MANUAL: mode_string = "Manual : Correction value is " + correction ; break; } MessageBox.Show(mode_string); } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>Dim mode As MBPLib1.LaserCalibrationTypes Dim dt As Date Dim manual As Long MBActX1.Equipment.SetLaserPowerCalibrationMode LASERCALIBRATION_AUTO, 0 MBActX1.Equipment.GetLaserPowerCalibrationMode mode, dt, manual Debug.Print mode, dt, manual</pre>			

SetLaserPowerCalibrationMode

Sets the laser power calibration mode.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	LaserCalibrationTypes	CalibrationType	LASERCALIBRATION_DEFAULTS	
			LASERCALIBRATION_AUTO	
			LASERCALIBRATION_MANUAL	
	out int	ManualCorrectionValue	Manual correction value	
Return value	-			
Note	If LaserCalibrationTypes is LASERCALIBRATION_DEFAULTS, 0 is sent to the unit regardless of the correction value. Exception: If the equipment is not connected with the unit The “Communication error” occurs. If out of input range The “Out of range error” occurs.			
Detailed example C#	MBPLib1.LaserCalibrationTypes mode = MBPLib1.LaserCalibrationTypes.LASERCALIBRATION_AUTO; int correction = 0; // 0 when LASERCALIBRATION_AUTO set try{ axMBActX1.Equipment.SetLaserPowerCalibrationMode(mode, correction); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			
Simple example VBA	See “MBActX1.Equipment.GetLaserPowerCalibrationMode”			

StartLaserPowerAutoCalibration

Starts the laser power auto-calibration.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvError EvExpansionStart/ End *1	Blocking communication	*2	
Argument	-			
Return value	-			
Note	The terminal block is in the “Auto-calibrating laser power” status during the execution, causing the EvError event to occur when starting and exiting. After normal termination, the expansion start and end events occur to return the setting to the current setting. Exception: If the equipment is not connected with the unit The “Communication error” occurs. * 1 Even if the StopLaserPowerAutoCalibration method is not used, the EvExpansionStart/End event occurs. * 2 Regardless of the value, True or False, of IsBlockingCommunication, the function exits before the communication ends.			
Detailed example C#	<pre>try{ axMBActX1.Equipment.StartLaserPowerAutoCalibration(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Equipment.StartLaserPowerAutoCalibration			

StopLaserPowerAutoCalibration

Stops the laser power auto-calibration.

Type	Method	Mode	Operation	✓
Memory allocation	-			Editing
Event	EvError EvExpansionStart/ End	Blocking communication	*1	
Argument	-			
Return value	-			
Note	The “Auto-calibration laser power” terminal block status is released when the execution is stopped, and error event occurs. After the execution is stopped, the expansion start and end events occur to return the setting to the current setting. Exception: If the equipment is not connected with the unit The “Communication error” occurs. * 1 Regardless of the value, True or False, of IsBlockingCommunication, the function exits before the communication ends.			
Detailed example C#	<pre>try{ axMBActX1.Equipment.StopLaserPowerAutoCalibration(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Equipment.StartLaserPowerAutoCalibration MBActX1.Equipment.StopLaserPowerAutoCalibration			

StartLaserPowerMaintenance

Starts laser power maintenance.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvMarkingStart/End *1 EvExpansionStart/ End	Blocking communication	*2	
Argument	double	Power	Laser power (%, The tenth place digit)	
	int	Frequency	Pulse frequency (kHz)	
	bool	IsOpenShutter	With/without shutter opening/ closing	
Return value	-			
Note	At the time the start of marking, the start making event occurs. Also, at the time of normal end of the marking, the marking complete event occurs. At the same time, the expansion starts to return the setting to the current setting. When this is completed, an expansion end event occurs. Exception: If the equipment is not connected with the unit The "Communication error" occurs. * 1 Even if the StopLaserPowerMaintenance method is not used, the EvMarkingStart/End or EvExpansionStart/End event occurs. * 2 Regardless of the value, True or False, of IsBlockingCommunication, the function exits before the communication ends.			
Example	See GetLaserPowerMaintenanceOutput.			

StopLaserPowerMaintenance

Stops laser power maintenance.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvMarkingEnd EvExpansionStart/ End	Blocking communication	*1	
Argument	-			
Return value	-			
Note	After the execution is stopped, a marking end event occurs. At the same time, the expansion starts to return the setting to the current setting. When this is completed, an expansion end event occurs. Exception: If the equipment is not connected with the unit The “Communication error” occurs. * 1 Regardless of the value, True or False, of IsBlockingCommunication, the function exits before the communication ends.			
Example	See GetLaserPowerMaintenanceOutput.			

GetLaserPowerMaintenanceOutput

Gets the laser power output.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	double	Output value (Unit: W) (The tenth place digit)		
Note	To get the output power during laser inspection, call this method regularly. The update cycle of the measured power is about 0.5 seconds, so calling the method at a short interval may cause you to get the same value continuously. Exception: If the equipment is not connected with the unit The "Communication error" occurs. Before executing StartLaserPowerMaintenanceMethod The "The laser output value could not be acquired" error occurs.			
Detailed example C#	<pre>double Power = 80.0; int Frequency = 100; bool IsOpenShutter = false; double Output; //Start try{ axMBActX1.Equipment.StartLaserPower Maintenance(Power, Frequency, IsOpenShutter); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); return; } //Measure the power 10 times at every second try{ for (int i=0; i < 10 ; i++) { Output = axMBActX1.Equipment.GetLaser PowerMaintenanceOutput(); //TODO Write the display processing here. System.Threading.Thread.Sleep(1000); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); } //End try{ axMBActX1.Equipment.StopLaserPowerMaintenance(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); } }</pre>			
Simple example VBA	MBActX1.Equipment.StartLaserPowerMaintenance 100, 0, False For i = 1 To 10 Application.Wait Now + TimeValue("00:00:01") Debug.Print MBActX1.Equipment. GetLaserPowerMaintenanceOutput Next i MBActX1.Equipment.StopLaserPowerMaintenance			

SetDistancePointerStatus

Sets the distance pointer to ON/OFF in the Operation status.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvError	Blocking communication	-	
Argument	-			
Return value	bool	IsLighting	True: Distance pointer ON False: Distance pointer OFF	
Note	The terminal block is in the “Distance pointer ON” status during the execution, causing the EvError event to occur when starting and exiting. Exception: If the equipment is not connected with the unit The “Communication error” occurs.			
Detailed example C#	<pre>bool IsLighting =true; try{ axMBActX1.Equipment.SetDistancePointerStatus (IsLighting); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Equipment.SetDistancePointerStatus False Debug.Print MBActX1.Equipment.GetDistancePointerStatus			

GetDistancePointerStatus

Gets the distance pointer status (ON/OFF) in the Operation status.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	bool	IsLighting	True: Distance pointer ON False: Distance pointer OFF	
Note	Exception: If the equipment is not connected with the unit The “Communication error” occurs.			
Detailed example C#	<pre>bool IsLighting; try{ IsLighting = axMBActX1.Equipment. DistancePointerStatus(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	See “SetDistancePointerStatus False”.			

SetCumulativeMarkingCount

Sets the cumulative marking count.

Type	Method	Mode	Operation	✓
Memory allocation	-			Editing
Event	-	Blocking communication	-	
Argument	-			
Return value	int	Number	Number of the cumulative marking count to set (0,1)	
	uint	Cumulative MarkingCount	Cumulative marking count (0 to 4294967295)	
Note	Exception: If the equipment is not connected with the unit The “Communication error” occurs. If the number of the cumulative marking count is out of input range The out of input range error occurs. If the cumulative marking count is out of input range The out of input range error occurs.			
Detailed example C#	int Number = 1; uint CumulativeMarkingCount = 0; try{ axMBActX1.Equipment.SetCumulativeMarkingCount(Number, CumulativeMarkingCount); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			

GetCumulativeMarkingCount

Gets the cumulative marking count.

Type	Method	Mode	Operation	✓
Memory allocation	-			Editing
Event	-	Blocking communication	-	
Argument	-			
Return value	int	Number	Number of the cumulative marking count to set (0,1)	
	uint	Cumulative MarkingCount	Cumulative marking count (0 to 4294967295)	
Note	Exception: If the equipment is not connected with the unit The “Communication error” occurs. If out of input range The out of input range error occurs.			
Detailed example C#	int Number = 1; uint CumulativeMarkingCount; try{ CumulativeMarkingCount = axMBActX1.Equipment. GetCumulativeMarkingCount(Number); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			

SetCumulativeMarkingCountString

Sets the cumulative marking count as a character string. (For VBA)

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	Int	Number	Number of the cumulative marking count to set (0,1)	
	string	Cumulative MarkingCount	Cumulative marking count (0 to 4294967295)	
Return value	-			
Note	VBA cannot handle unsigned 32-bit integer; therefore, use this method instead of SetCumulativeMarkingCount. Exception: If the equipment is not connected with the unit The “Communication error” occurs. If the number of the cumulative marking count is out of input range The out of input range error occurs. If the cumulative marking count is out of input range The out of input range error occurs.			
Simple example VBA	MBActX1.Equipment.SetCumulativeMarkingCountString 0, 50 Debug.Print MBActX1.Equipment.GetCumulativeMarkingCountString(0)			

GetCumulativeMarkingCountString

Gets the cumulative marking count as a character string. (For VBA)

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	Number	Number of the cumulative marking count to set (0,1)	
	string	Cumulative MarkingCount	Cumulative marking count (0 to 4294967295// Character string)	
Return value	-			
Note	VBA cannot handle unsigned 32-bit integer; therefore, use this method instead of GetCumulativeMarkingCount. Exception: If the equipment is not connected with the unit The “Communication error” occurs. If out of input range The out of input range error occurs.			
Simple example VBA	See SetCumulativeMarkingCountString.			

ApplyLaserScanningCalibration

Updates the correction status of the controller with the correction value setting file created by Laser Scanning Calibrator Plus.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	string	FileName	Correction value setting file name saved in the PC	
Return value	-			
Note	Specify the path of correction value setting file to transfer. The absolute path can also be specified. When only the file name is specified, use the file in the Calibration data folder of Marking Builder Plus. (C:\Users\{User name}\Documents\Marking Builder Plus\Calibration As with the case of the Laser Scanning Calibrator Plus, executing this method retains the corrected status even when the power is once turned off and turned on again. To reset to the default values, use the Laser Scanning Calibrator Plus. Exception: If the equipment is not connected with the unit The "Communication error" occurs. If a file without its extension is specified The message, "Failed to open transmission file", occurs. If LaserScanningCalibrator is being started The message, "Failed to correct the value (19)," occurs.			
Detailed example C#	<pre>try{ axMBActX1.Equipment.ApplyLaserScanningCalibration("Controller1.LSCP2"); } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Equipment.ApplyLaserScanningCalibration "test.lscp2"			

GetOperatingTime

Gets the operation time of the controller.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	int	Date and time of the controller (H)		
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>int ControllerOperatingTime; try{ ControllerOperatingTime = axMBActX1.Equipment.GetOperatingTime(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Equipment.GetOperatingTime			

GetScannerOperatingTime

Gets the operation time of the scanner.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	int	Date and time of the scanner (H)		
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>int ScannerOperatingTime; try{ ScannerOperatingTime = axMBActX1.Equipment.GetScannerOperatingTime(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Equipment.GetScannerOperationTime			

GetShutterOperatingCount

Gets the number of shutter cycles.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	int	Number of times the shutter has been operated.		
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>int ShutterOperatingCount; try{ ShutterOperatingCount = axMBActX1.Equipment. GetShutterOperatingCount(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Equipment.GetShutterOperatingCount			

GetLaserSafetyModuleOperatingCount

Gets the number of laser safety module cycles.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	int	count	Number of times the laser safety module has been operated.	
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>int count; try{ count = axMBActX1.Equipment.GetLaserSafetyModuleOperatingCount(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>Dim count As Integer count = MBActX1.Equipment.GetLaserSafetyModuleOperatingCount</pre>			

GetMarkingUnitTemperature

Gets the head temperature.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	double	Head temperature (°C)		
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>double MarkingUnitTemperature; try{ MarkingUnitTemperature = axMBActX1.Equipment. GetMarkingUnitTemperature(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Equipment. GetMarkingUnitTemperature			

GetControllerTemperature

Gets the controller temperature.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	double	Controller temperature (°C)		
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>double ControllerTemperature; try{ ControllerTemperature = axMBActX1.Equipment. GetControllerTemperature(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Equipment.GetControllerTemperature			

GetLaserPowerCalibrationResult

Gets the results of the laser power calibration.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	double	Results of the laser power calibration		
Note	Exception: If the equipment is not connected with the unit The “Communication error” occurs.			
Detailed example C#	<pre>double LaserPowerCalibrationResult; try{ LaserPowerCalibrationResult = axMBActX1.Equipment. GetLaserPowerCalibrationResult(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Equipment. GetLaserPowerCalibrationResult			

GetLaserAccumulatedTime

Gets the time that the laser has been in the excited state.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	int	Excited laser time (unit: hour)		
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>int LaserAccumulatedTime; try{ LaserAccumulatedTime = axMBActX1.Equipment. GetLaserAccumulatedTime(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Equipment. GetLaserAccumulatedTime			

GetWindowCheckResult

Gets the final execution result of the lens inspection.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	out bool	result	Existence of a result	
	out int	year	Year conducted	
	out int	month	Month conducted	
	out int	date	Date conducted	
	out int	value	Conducted result (0 to 100)	
Return value	-			
Note	If the equipment is not connected with the unit "Not connected" occurs. When the lens inspection is being executed (only the lens inspection at startup) "The lens inspection is being executed" occurs.			
Supplement	-			
Example	Dim result As Boolean Dim year As Integer Dim month As Integer Dim day As Integer Dim value As Integer Call MBActX1.Equipment.GetWindowCheckResult(result, year, month, day, value)			

GetStartupWindowCheckEnabled

Gets whether or not to enable the lens inspection at startup.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	bool		Enabled/Disabled	
Note	If the equipment is not connected with the unit "Not connected" occurs.			
Supplement	-			
Example	Dim windowCheckEnabled As Boolean windowCheckEnabled = MBActX1.Equipment.GetStartupWindowCheckEnabled			

SetStartupWindowCheckEnabled

Sets whether or not to enable the lens inspection at startup.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	✓	
Argument	bool	windowCheck Enabled	Enabled/Disabled	
Return value	-			
Note	If the equipment is not connected with the unit "Not connected" occurs.			
Supplement	-			
Example	(lens inspection disabled) Call MBActX1.Equipment.SetStartupWindowCheckEnabled(False) (lens inspection enabled) Call MBActX1.Equipment.SetStartupWindowCheckEnabled(True)			

GetWindowCheckAlarmThreshold

Gets the warning threshold of the lens inspection.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	Int		Threshold	
Note	If the equipment is not connected with the unit "Not connected" occurs.			
Supplement	-			
Example	Dim threshold As Integer threshold = MBActX1.Equipment.GeWindowCheckAlarmThreshold			

SetWindowCheckAlarmThreshold

Sets the warning threshold of the lens inspection. (When the threshold is 0, it becomes the warning invalid operation)

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	✓	
Argument	Int	threshold	Threshold	
Return value	-			
Note	If the equipment is not connected with the unit "Not connected" occurs. If the other device is in the status of acquiring mastership "Priority error" occurs.			
Supplement	-			
Example	Call MBActX1.Equipment.SetWindowCheckAlarmThreshold(10)			

GetWindowCheckSensitivity

Gets the sensitivity adjustment of the lens inspection.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	WindowCheckSensitivityTypes	Sensitivity adjustment		
Note	If the equipment is not connected with the unit "Not connected" occurs.			
Supplement	-			
Example	Dim SensitivityType As WindowCheckSensitivityTypes SensitivityType = MBActX1.Equipment.GetWindowCheckSensitivity			

SetWindowCheckSensitivity

Sets the sensitivity adjustment of the lens inspection at startup.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	WindowCheckSensitivityTypes	sensitivityType	Sensitivity adjustment	
Note	If the equipment is not connected with the unit “Not connected” occurs.			
Supplement	-			
Example	Call MBActX1.Equipment.SetWindowCheckSensitivity (WINDOW_CHECK_SENSITIVITY_HIGH2)			

StartWindowCheck

Executes the lens inspection.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	-			
Note	If the equipment is not connected with the unit “Not connected” occurs. If the other device is in the status of acquiring mastership “Priority error” occurs.			
Supplement	-			
Example	Call MBActX1.Equipment.StartWindowCheck			

Property

LaserMarkerVersion

Gets the version number of the laser marker.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	string	Character string of the version number		
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	string Version = axMBActX1.Equipment.LaserMarkerVersion;			
Simple example VBA	Debug.Print MBActX1.Equipment.LaserMarkerVersion			

ControllerSerialNo

Gets the controller serial No.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	Controller serial No.		
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	int Version = axMBActX1.Equipment.ControllerSerialNo;			
Simple example VBA	Debug.Print MBActX1.Equipment.ControllerSerialNo			

HeadSerialNo

Gets the head serial No.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	Head serial No.		
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	int Version = axMBActX1.Equipment.HeadSerialNo;			
Simple example VBA	Debug.Print MBActX1.Equipment.HeadSerialNo			

MBMatrix

Method

MatrixCell

Gets the individual cell setting of a matrix.

Type	Method	Mode	Operation	-
Memory allocation	✓		Editing	✓
Event	-	Blocking communication	-	
Argument	int	Row	Row	
	int	Column	Column	
Return value	MBMatrixCell	-	Cell individual setting object	
Note	Exception: If the number of rows/columns is out of input range The out of input range error occurs.			
Detailed example C#	<pre>int row = 2; int column = 3; try{ if(axMBAcX1.Matrix(0) != null){ axMBAcX1.Matrix(0).MatrixCell(row,column). OffsetX = 5.000; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>MBAcX1.Matrix(0).MatrixCell(1, 1).OffsetX = 0.5 MBAcX1.Matrix(0).MatrixCell(1, 1).OffsetY = 0.5 MBAcX1.Matrix(0).MatrixCell(1, 1).OffsetZ = 0.5 Debug.Print MBAcX1.Matrix(0).MatrixCell(1, 1).OffsetX, MBAcX1.Matrix(0).MatrixCell(1, 1).OffsetY, MBAcX1.Matrix(0).MatrixCell(1, 1).OffsetZ</pre>			

SetWidth

Sets the width of a matrix.

Type	Method	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	MatrixSizeTypes	MatrixSizeType	MATRIXSIZE_ CELLSIZE	
	//Cell size	Column	Column	
Return value	-			
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>try{ if(axMBAcX1.Matrix(0) != null){ axMBAcX1.Matrix(0).SetWidth(MatrixSizeType. MATRIXSIZE_ CELLSIZE, 5.000); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBAcX1.Matrix(0).SetWidth MATRIXSIZE_ FULLSIZE, 80			

SetHeight

Sets the height of a matrix.

Type	Method	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	MatrixSizeTypes	MatrixSizeType	MATRIXSIZE_CELLSIZE //Cell size MATRIXSIZE_DISTANCE //Distance between the centers of the cells at both ends MATRIXSIZE_FULLSIZE //Size of the entire matrix	
	double	Height	Height (mm)	
Return value	-			
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>try{ if(axMBAcX1.Matrix(0) != null){ axMBAcX1.Matrix(0).SetHeight(MatrixSizeType.MATRIXSIZE_CELLSIZE, 5.000); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBAcX1.Matrix(0).SetHeight MATRIXSIZE_FULLSIZE, 80			

Property

RowCount

Gets the number of rows in a matrix.

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Number of rows		
Significant figures	-			
Note	-			
Detailed example C#	<pre>try{ if(10 < axMBAcX1.Matrix.RowCount); //TODO Write the processing to be performed if the condition is true. }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBAcX1.Matrix(0).RowCount			

ColumnCount

Gets the number of columns of a matrix.

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Number of columns		
Significant figures	-			
Note	-			
Detailed example C#	<pre>try{ if(10 < axMBAcX1.Matrix.ColumnCount); //TODO Write the processing to be performed if the condition is true. }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBAcX1.Matrix(0).ColumnCount			

X

Gets/sets the reference X coordinate.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Reference X coordinate		
Significant figures	3rd decimal place			
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>try{ axMBAcX1.Matrix.X = 10.000; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBAcX1.Matrix(0).X			

Y

Gets/sets the reference Y coordinate.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Reference Y coordinate		
Significant figures	3rd decimal place			
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>try{ axMBAcX1.Matrix.Y = 10.000; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBAcX1.Matrix(0).Y			

CellWidth

Gets the width of one cell in a matrix.

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Width of one cell in a matrix		
Note	-			
Detailed example C#	<pre>try{ if(axMBAcX1.Matrix(0) != null){ double width = axMBAcX1.Matrix(0).CellWidth; double height = axMBAcX1.Matrix(0).CellHeight; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBAcX1.Matrix(0).CellWidth Debug.Print MBAcX1.Matrix(0).CellHeight			

CellHeight

Gets the height of one cell in a matrix.

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Height of one cell in a matrix		
Note	-			
Example	See CellWidth.			

MatrixWidth

Gets the entire width of a matrix.

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Entire width of a matrix		
Note	-			
Detailed example C#	<pre>try{ if(axMBActX1.Matrix(0) != null){ double width = axMBActX1.Matrix(0).MatrixWidth; double height = axMBActX1.Matrix(0).MatrixHeight; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Matrix(0).MatrixWidth Debug.Print MBActX1.Matrix(0).MatrixHeight			

MatrixHeight

Gets the entire height of a matrix.

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Entire height of a matrix		
Note	-			
Example	See MatrixWidth.			

MBMatrixCell

Property

OffsetX

Gets/sets the offset value of the X coordinate of an individual cell.

Type	Property (RW)	Mode	Operation	-	Type	double	Reference X coordinate
Memory allocation	-		Editing	✓	Significant figures	3rd decimal place	
Event	-	Blocking communication	-				
Note	Exception: If out of input range The out of input range error occurs.						
Detailed example C#	<pre>try{ //Offset all the elements. for(int row=1; row <= axMBActX1.Matrix.RowCount; row++){ for(int col=1; col <= axMBActX1.Matrix.ColumnCount; col++){ axMBActX1.MatrixCell(row, col).OffsetX = 1.000; } } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>						
Simple example VBA	See "MatrixCell".						

OffsetY

Gets/sets the offset value of the Y coordinate of an individual cell.

Type	Property (RW)	Mode	Operation	-	Type	double	Reference Y coordinate
Memory allocation	-		Editing	✓	Significant figures	3rd decimal place	
Event	-	Blocking communication	-				
Note	Exception: If out of input range The out of input range error occurs.						
Detailed example C#	<pre>try{ //Offset all the elements. for(int row=1; row <= axMBActX1.Matrix.RowCount; row++){ for(int col=1; col <= axMBActX1.Matrix.ColumnCount; col++){ axMBActX1.MatrixCell(row, col).OffsetY = 1.000; } } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>						
Simple example VBA	See "MatrixCell".						

OffsetZ

Gets/sets the offset value of the Z coordinate of an individual cell.

Type	Property (RW)	Mode	Operation	-	Type	double	Reference Z coordinate
Memory allocation	-		Editing	✓	Significant figures	3rd decimal place	
Event	-	Blocking communication	-				
Note	Exception: If out of input range The out of input range error occurs.						
Detailed example C#	<pre>try{ //Offset all the elements. for(int row=1; row <= axMBActX1.Matrix.RowCount; row++){ for(int col=1; col <= axMBActX1.Matrix.ColumnCount; col++){ axMBActX1.MatrixCell(row, col).OffsetZ = 1.00; } } } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>						
Simple example VBA	See "MatrixCell".						

OffsetAngle

Gets/sets the offset value of the θ angle of an individual cell.

Type	Property (RW)	Mode	Operation	-	Type	double	Angle
Memory allocation	-		Editing	✓	Significant figures	3rd decimal place	
Event	-	Blocking communication	-				
Note	Exception: If out of input range The out of input range error occurs.						
Detailed example C#	<pre>try{ //Offset all the elements. for(int row=1; row <= axMBActX1.Matrix.RowCount; row++){ for(int col=1; col <= axMBActX1.Matrix.ColumnCount; col++){ axMBActX1.MatrixCell(row, col).OffsetAngle = 10.00; } } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>						
Simple example VBA	MBActX1.Matrix(0).MatrixCell(1, 1).OffsetAngle = 20 Debug.Print MBActX1.Matrix(0).MatrixCell(1, 1).OffsetAngle						

IsMarkingEnable

Gets/sets whether marking is enabled for an individual cell.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	bool	True: Enable the marking. False: Disable the marking.		
Note	-			
Detailed example C#	<pre>try{ //Disable the marking of the element in 2nd row and 3rd column. axMBActX1.MatrixCell(2, 3).IsMarkingEnable = false; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Matrix(0).MatrixCell(1, 1).IsMarkingEnable = True Debug.Print MBActX1.Matrix(0).MatrixCell(1, 1).IsMarkingEnable			

MarkingDelayTime

Gets/sets the wait time for start marking of an individual cell.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Wait time for start cell marking (ms)		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>try{ int Row = axMBActX1.Matrix(0).RowCount; for(int i=0; i<Row; i++){ axMBActX1.Matrix(0).MatrixCell(i, 0). MarkingDelayTime = 10.0; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Matrix(0).MatrixCell(1, 1).MarkingDelayTime = 0.1 Debug.Print MBActX1.Matrix(0).MatrixCell(1, 1).MarkingDelayTime			

OriginX

Gets the X coordinate of the point of origin in a cell.

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	X coordinate of the point of origin in a cell		
Note	The offset X, Y, and θ angles in the cell individual setting are not considered.			
Detailed example C#	<pre>int row = 2; int column = 3; try{ if(axMBActX1.Matrix(0) != null){ double x = axMBActX1.Matrix(0). MatrixCell(row,column).OriginX; double y = axMBActX1.Matrix(0). MatrixCell(row,column).OriginY; axMBActX1.Operation.SetCameraPosition(x, y, 0, true); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Matrix(0).MatrixCell(1, 1).OriginX Debug.Print MBActX1.Matrix(0).MatrixCell(1, 1).OriginY			

OriginY

Gets the Y coordinate of the point of origin in a cell.

Type	Property(R)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Y coordinate of the point of origin in a cell		
Note	The offset X, Y, and θ angles in the cell individual setting are not considered.			
Example	See OriginX.			

MBOperation

Method

SetCurrentCounterValue

Sets the current value of the counter.

Type	Method	Mode	Operation	✓		
Memory allocation	-		Editing	✓		
Event	EvExpansionStart/End	Blocking communication	✓			
Argument	int	JobNo	Setting number			
	CounterNoTypes	CounterNo	COUNTERNO_0	COUNTERNO_5	COUNTERNO_A	COUNTERNO_F
			COUNTERNO_1	COUNTERNO_6	COUNTERNO_B	COUNTERNO_G
			COUNTERNO_2	COUNTERNO_7	COUNTERNO_C	COUNTERNO_H
			COUNTERNO_3	COUNTERNO_8	COUNTERNO_D	COUNTERNO_I
			COUNTERNO_4	COUNTERNO_9	COUNTERNO_E	COUNTERNO_J
uint *1	CounterValue	Value to change				
Return value	-					
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the program number is out of input range The out of input range error occurs. If the counter value is out of input range The out of input range error occurs. * 1 VBA cannot handle unsigned 32-bit integer; therefore, use SetCurrentCounterString instead of this method.					
Detailed example C#	int JobNo = 10; MBPLib1.CounterNoTypes CounterNo = MBPLib1.CounterNoTypes.COUNTERNO_5; uint CurrentCounterValue = 10000; try{ axMBActX1.Operation.SetCurrentCounterValue(JobNo, CounterNo, CurrentCounterValue); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }					

GetCurrentCounterValue

Gets the current value of the counter.

Type	Method	Mode	Operation	✓				
Memory allocation	-		Editing	✓				
Event	-	Blocking communication	-					
Argument	int	JobNo	Setting number					
	CounterNoTypes	CounterNo	COUNTERNO_0	COUNTERNO_5	COUNTERNO_A	COUNTERNO_F		
			COUNTERNO_1	COUNTERNO_6	COUNTERNO_B	COUNTERNO_G		
			COUNTERNO_2	COUNTERNO_7	COUNTERNO_C	COUNTERNO_H		
			COUNTERNO_3	COUNTERNO_8	COUNTERNO_D	COUNTERNO_I		
			COUNTERNO_4	COUNTERNO_9	COUNTERNO_E	COUNTERNO_J		
Return value	uint *1	-	Counter value					
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the program number is out of input range The out of input range error occurs. * 1 VBA cannot handle unsigned 32-bit integer; therefore, use GetCurrentCounterString instead of this method.							
Detailed example C#	<pre>int JobNo = 10; MBPLib1.CounterNoTypes CounterNo = MBPLib1.CounterNoTypes.COUNTERNO_5; uint CurrentCounterValue; try{ CurrentCounterValue = axMBActX1.Operation.GetCurrentCounterValue(JobNo, CounterNo); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>							

SetCurrentCounterString

Sets the current value of the counter as a character string. (For VBA)

Type	Method	Mode	Operation	✓		
Memory allocation	-		Editing	✓		
Event	EvExpansionStart/End	Blocking communication	✓			
Argument	int	JobNo	Setting number			
	CounterNoTypes	CounterNo	COUNTERNO_0	COUNTERNO_5	COUNTERNO_A	COUNTERNO_F
			COUNTERNO_1	COUNTERNO_6	COUNTERNO_B	COUNTERNO_G
			COUNTERNO_2	COUNTERNO_7	COUNTERNO_C	COUNTERNO_H
			COUNTERNO_3	COUNTERNO_8	COUNTERNO_D	COUNTERNO_I
			COUNTERNO_4	COUNTERNO_9	COUNTERNO_E	COUNTERNO_J
string	CounterValue	Counter value (character string)				
Return value	-					
Note	VBA cannot handle unsigned 32-bit integer; therefore, use this method instead of SetCurrentCounterValue. Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the program number is out of input range The out of input range error occurs. If the counter value is out of input range The out of input range error occurs.					
Simple example VBA	MBActX1.Operation.SetCurrentCounterString 0, COUNTERNO_0, 8 Debug.Print MBActX1.Operation.GetCurrentCounterString(0, COUNTERNO_0)					

GetCurrentCounterString

Gets the current value of the counter as a character string. (For VBA)

Type	Method	Mode	Operation	✓			
Memory allocation	-		Editing	✓			
Event	-	Blocking communication	-				
Argument	int	JobNo	Setting number				
	CounterNoTypes	CounterNo	COUNTERNO_0	COUNTERNO_5	COUNTERNO_A	COUNTERNO_F	
			COUNTERNO_1	COUNTERNO_6	COUNTERNO_B	COUNTERNO_G	
			COUNTERNO_2	COUNTERNO_7	COUNTERNO_C	COUNTERNO_H	
			COUNTERNO_3	COUNTERNO_8	COUNTERNO_D	COUNTERNO_I	
			COUNTERNO_4	COUNTERNO_9	COUNTERNO_E	COUNTERNO_J	
Return value	string	-	Counter value (character string)				
Note	VBA cannot handle unsigned 32-bit integer; therefore, use this method instead of GetCurrentCounterValue. Exception: If the equipment is not connected with the unit The “Communication error” occurs. If the program number is out of input range The out of input range error occurs.						
Detailed example C#	See “SetCurrentCounterString”.						

SetCurrentCounterRepetition

Sets the repetition value of a counter.

Type	Method	Mode	Operation	✓		
Memory allocation	-		Editing	✓		
Event	EvExpansionStart/End	Blocking communication	✓			
Argument	int	JobNo	Setting number			
	CounterNoTypes	CounterNo	COUNTERNO_0	COUNTERNO_5	COUNTERNO_A	COUNTERNO_F
			COUNTERNO_1	COUNTERNO_6	COUNTERNO_B	COUNTERNO_G
			COUNTERNO_2	COUNTERNO_7	COUNTERNO_C	COUNTERNO_H
			COUNTERNO_3	COUNTERNO_8	COUNTERNO_D	COUNTERNO_I
			COUNTERNO_4	COUNTERNO_9	COUNTERNO_E	COUNTERNO_J
uint	CurrentCounter Repetition	Current repetitions				
Return value	-					
Note	Exception: If the equipment is not connected with the unit The “Communication error” occurs. If the current repetitions is out of input range The out of input range error occurs. If the program number is out of input range The out of input range error occurs.					
Detailed example C#	<pre>int JobNo = 10; MBPLib1.CounterNoTypes CounterNo = MBPLib1.CounterNoTypes.COUNTERNO_5; uint CurrentCounterRepetition = 10000; try{ axMBActX1.Operation.SetCurrentCounterRepetition(JobNo, CounterNo, CurrentCounterRepetition); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>					

GetCurrentCounterRepetition

Gets the repetition value of a counter.

Type	Method	Mode	Operation	✓			
Memory allocation	-		Editing	✓			
Event	-	Blocking communication	-				
Argument	int	JobNo	Setting number				
	CounterNoTypes	CounterNo	COUNTERNO_0	COUNTERNO_5	COUNTERNO_A	COUNTERNO_F	
			COUNTERNO_1	COUNTERNO_6	COUNTERNO_B	COUNTERNO_G	
			COUNTERNO_2	COUNTERNO_7	COUNTERNO_C	COUNTERNO_H	
			COUNTERNO_3	COUNTERNO_8	COUNTERNO_D	COUNTERNO_I	
			COUNTERNO_4	COUNTERNO_9	COUNTERNO_E	COUNTERNO_J	
uint	CurrentCounter Repetition	Current repetitions					
Return value	-						
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the program number is out of input range The out of input range error occurs.						
Detailed example C#	<pre>int JobNo = 10; MBPLib1.CounterNoTypes CounterNo = MBPLib1.CounterNoTypes.COUNTERNO_5; uint CurrentCounterRepetition; try{ CurrentCounterRepetition = axMBActX1.Operation.GetCurrentCounterRepetition(JobNo, CounterNo); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>						

SetCurrentCounterRepetitionString

Sets the current repetitions of the counter as a character string. (For VBA)

Type	Method	Mode	Operation	✓				
Memory allocation	-		Editing	✓				
Event	EvExpansionStart/End	Blocking communication	✓					
Argument	int	JobNo	Setting number					
	CounterNoTypes	CounterNo	COUNTERNO_0	COUNTERNO_5	COUNTERNO_A	COUNTERNO_F		
			COUNTERNO_1	COUNTERNO_6	COUNTERNO_B	COUNTERNO_G		
			COUNTERNO_2	COUNTERNO_7	COUNTERNO_C	COUNTERNO_H		
			COUNTERNO_3	COUNTERNO_8	COUNTERNO_D	COUNTERNO_I		
			COUNTERNO_4	COUNTERNO_9	COUNTERNO_E	COUNTERNO_J		
	string	CurrentCounter Repetition	Current repetitions					
Return value	-							
Note	VBA cannot handle unsigned 32-bit integer; therefore, use this method instead of SetCurrentCounterRepetition. Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the current repetitions is out of input range The out of input range error occurs. If the program number is out of input range The out of input range error occurs.							
Simple example VBA	MBActX1.Operation.SetCurrentCounterRepetitionString 0, COUNTERNO_0, 4 Debug.Print MBActX1.Operation.GetCurrentCounterRepetitionString(0, COUNTERNO_0)							

GetCurrentCounterRepetitionString

Gets the current repetitions of the counter as a character string. (For VBA)

Type	Method	Mode	Operation	✓				
Memory allocation	-		Editing	✓				
Event	-	Blocking communication	-					
Argument	int	JobNo	Setting number					
	CounterNoTypes	CounterNo	COUNTERNO_0	COUNTERNO_5	COUNTERNO_A	COUNTERNO_F		
			COUNTERNO_1	COUNTERNO_6	COUNTERNO_B	COUNTERNO_G		
			COUNTERNO_2	COUNTERNO_7	COUNTERNO_C	COUNTERNO_H		
			COUNTERNO_3	COUNTERNO_8	COUNTERNO_D	COUNTERNO_I		
			COUNTERNO_4	COUNTERNO_9	COUNTERNO_E	COUNTERNO_J		
Return value	string	CurrentCounter Repetition	Current repetitions					
Note	VBA cannot handle unsigned 32-bit integer; therefore, use this method instead of GetCurrentCounterRepetition. Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the program number is out of input range The out of input range error occurs.							
Simple example VBA	See SetCurrentCounterRepetitionString.							

GetJobTitle

Gets a job title.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	JobNo	Setting number	
Return value	string	Title name		
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the program number is out of input range The out of input range error occurs.			
Detailed example C#	<pre>int JobNo = 10; string JobTitle; try{ JobTitle = axMBActX1.Operation.GetJobTitle(JobNo); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Operation.GetJobTitle(0)			

GetWorkDistanceGap

Gets the working distance.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	double	Difference between the working distance and the reference distance (mm)		
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. For an unsupported model The message, "S095: Model limitation error", occurs. If the measurement failed (Marking/2D code reader is now reading/Finder is being operated/Unexcited laser state) An error occurs.			
Detailed example C#	double workdistance; try{ workdistance = axMBActX1.Operation. GetWorkDistanceGap(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			
Simple example VBA	Debug.Print MBActX1.Operation.GetWorkDistanceGap			

SetCharString2

Sets the character string.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvExpansionStart/End *1	Blocking communication	✓	
Argument	int	JobNo	Setting number	
	int	BlockNo	Block number	
	string	CharString	Character string	
Return value	-			
Note	In the case of a logo block, specify a logo file name which has been saved in the controller in advance. Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the program number is out of input range The out of input range error occurs. If the block number is out of input range The out of input range error occurs. If the character string is out of input range The out of input range error occurs. * 1 If the BeginTransaction method is used, the EvExpansionStart/End event does not occur.			
Detailed example C#	<pre>int JobNo = 10; int BlockNo = 2; string CharString = "ABC"; try{ axMBActX1.Operation.SetCharString2(JobNo, BlockNo, CharString); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Operation.SetCharString2 0, 0, "ActiveX" Debug.Print MBActX1.Operation.GetCharString2(0, 0)			

GetCharString2

Gets the character string.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	JobNo	Setting number	
	int	BlockNo	Block number	
Return value	string		Character string	
Note	For a logo block, gets the logo file name. Exception: If the equipment is not connected with the unit The “Communication error” occurs. If the program number is out of input range The out of input range error occurs. If the block number is out of input range The out of input range error occurs.			
Detailed example C#	<pre>int JobNo = 10; int BlockNo = 2; string CharString; try{ CharString = axMBActX1.Operation. GetCharString2(JobNo, BlockNo); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Operation.GetCharString2(0, 0)			

SetWorkpiecePositionAdjustment

Sets the workpiece position adjustment.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvExpansionStart/ End *1	Blocking communication	✓	
Argument	int	JobNo	Setting number	
	double	ReferenceX	Movement reference point (X) (mm, the thousandth place digit)	
	double	ReferenceY	Movement reference point (Y) (mm, the thousandth place digit)	
	double	OffsetX	Correction amount (X) (mm, the thousandth place digit)	
	double	OffsetY	Correction amount (Y) (mm, the thousandth place digit)	
	double	OffsetAngle	Correction amount θ angle (°, The thousandth place digit)	
Return value	-			
Note	Exception: If the equipment is not connected with the unit The “Communication error” occurs. If the program number is out of input range The out of input range error occurs. If the movement reference point or Correction amount is out of input range The out of input range error occurs. * 1 If the BeginTransaction method is used, the EvExpansionStart/End event does not occur.			
Detailed example C#	int JobNo = 10; double ReferenceX = 1.000; double ReferenceY = 2.000; double OffsetX = 3.000; double OffsetY = 4.000; double OffsetAngle = 5.000; try{ axMBActX1.Operation.SetWorkpiecePositionAdjustment (JobNo, ReferenceX, ReferenceY, OffsetX, OffsetY, OffsetAngle); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			
Simple example VBA	Dim ProgramNo As Integer Dim ReferenceX As Double Dim ReferenceY As Double Dim OffsetX As Double Dim OffsetY As Double Dim OffsetAngle As Double MBActX1.Operation.SetWorkpiecePositionAdjustment 0, 0, 0, 10, 10, 45 MBActX1.Operation.GetWorkpiecePositionAdjustment JobNo, ReferenceX, ReferenceY, OffsetX, OffsetY, OffsetAngle Debug.Print JobNo Debug.Print ReferenceX Debug.Print ReferenceY Debug.Print OffsetX Debug.Print OffsetY Debug.Print OffsetAngle			

GetWorkpiecePositionAdjustment

Gets the workpiece position adjustment.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	JobNo	Setting number	
	out double	ReferenceX	Movement reference point (X) (mm, the thousandth place digit)	
	out double	ReferenceY	Movement reference point (Y) (mm, the thousandth place digit)	
	out double	OffsetX	Correction amount (X) (mm, the thousandth place digit)	
	out double	OffsetY	Correction amount (Y) (mm, the thousandth place digit)	
	out double	OffsetAngle	Correction amount θ angle (°, The thousandth place digit)	
Return value	-			
Note	Exception: If the equipment is not connected with the unit The “Communication error” occurs. If the program number is out of input range The out of input range error occurs.			
Detailed example C#	<pre>int JobNo = 10; double ReferenceX; double ReferenceY; double OffsetX; double OffsetY; double OffsetAngle; try{ axMBActX1.Operation.GetWorkpiecePositionAdjustment (JobNo, out ReferenceX, out ReferenceY, out OffsetX, out OffsetY, out OffsetAngle); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	See “GetWorkpiecePositionAdjustment”.			

SetWorkpiecePositionAdjustmentZ

Sets the correction value in the height direction of workpiece position adjustment.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvExpansionStart/ End *1	Blocking communication	✓	
Argument	int	JobNo	Setting number	
	double	OffsetZ	Correction value in the height direction (mm, The thousandth place digit)	
Return value	-			
Note	This method is enabled only when the correction method in the height direction is "Fixed". Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the program number is out of input range The out of input range error occurs. If the correction value in the height direction is out of input range The out of input range error occurs. If the correction method in the height direction is other than "Fixed" An "unsupported error" occurs. * 1 If the BeginTransaction method is used, the EvExpansionStart/End event does not occur.			
Detailed example C#	<pre>int JobNo = 10; double OffsetZ = 5; try{ axMBActX1.Operation.SetWorkpiecePositionAdjustmentZ (JobNo, OffsetZ); } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Operation.SetWorkpiecePositionAdjustmentZ 0, 5 Debug.Print MBActX1.Operation.GetWorkpiecePosition AdjustmentZ(0)			

StartMarking

Starts the marking laser.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvMarkingStart/End *1	Blocking communication		✓
Argument	-			
Return value	-			
Note	At the time the start of marking, the start making event occurs. Also, at the time of normal end of the marking, the marking complete event occurs. Exception: If the equipment is not connected with the unit The "Communication error" occurs. * 1 Even if the StopMarking method is not used, the marking end event will occur after this making is completed.			
Detailed example C#	<pre>try{ axMBActX1.Operation.StartMarking(); } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>MBActX1.Operation.StartMarking</pre>			

GetWorkpiecePositionAdjustmentZ

Gets the correction value in the Z direction of workpiece position adjustment.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	Int	JobNo	Setting number	
	double	-	Correction value in the height direction (mm)	
Return value	-			
Note	This method is enabled only when the correction method in the height direction is "Fixed". Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the program number is out of input range The out of input range error occurs. If the correction method in the height direction is other than "Fixed" An "unsupported error" occurs.			
Detailed example C#	int JobNo = 10; double OffsetZ; try{ OffsetZ = axMBAActX1.Operation.GetWorkpiecePositionAdjustmentZ(JobNo); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			
Simple example VBA	See "SetWorkpiecePositionAdjustmentZ".			

StartGuideLaserMarking

Starts guide laser marking.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvMarkingStart/End *1	Blocking communication	✓	
Argument	GuideLaseTypes	GuideLaserType	GUIDELASER_ONETIME //Once	
			GUIDELASER_CONTINUOUS //Continuousness	
			GUIDELASER_AREA //Marking area	
			GUIDELASER_BLOCKFRAME //Block frame	
			GUIDELASER_WORKPIECE //Workpiece image	
Return value	-			
Note	The guide laser marking (Once) ends after the same marking time as that of the actual marking has passed. The other markings automatically end after about 30 seconds. To forcibly end the marking, use MethodStopMarking. During the guide laser marking, the shutter is closed. Unlike “StartMarking()”, this method exits when the guide laser marking starts and the status becomes the non-blocking communication (The setting of IsBlockingCommunication is disabled.). Exception: If the equipment is not connected with the unit The “Communication error” occurs. * 1 Even if the StopMarking method is not used, the marking end event will occur after this making is completed.			
Detailed example C#	try{ axMBActX1.Operation.StartGuideLaserMarking(MBPLib1.GuideLaserTypes.GUIDELASER_CONTINUOUS); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			
Simple example VBA	MBActX1.Operation.StartGuideLaserMarking GUIDELASER_CONTINUOUS			

StopMarking

Stops marking.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvMarkingEnd	Blocking communication	-	
Argument	-			
Return value	-			
Note	When the block communication is ON (MBActX1.IsBlockingCommunication = True), StopMarking cannot be interrupted during marking. When using StopMarking to stop marking, combine the jobs so that it becomes MBActX1.IsBlockingCommunication = False during the marking. When this method is executed during normal marking (StartMarking, terminal block), EvMarkingEndEvent is sent after the marking is cancelled. When this method is executed during guide laser marking (StartGuideLaserMarking), the EvMarkingEnd even occurs after marking is cancelled. Exception: If the equipment is not connected with the unit The “Communication error” occurs.			
Detailed example C#	try{ axMBActX1.Operation.StopMarking(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			
Simple example VBA	MBActX1.Operation.StopMarking			

GetCurrentJobNo

Gets the current program number running on the unit.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	int	Current program number		
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>int JobNo; try{ JobNo = axMBActX1.Operation.GetCurrentJobNo(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Operation.SetCurrentJobNo 0 Debug.Print MBActX1.Operation.GetCurrentJobNo			

ClearError

Clears the error status in the unit.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvError *1	Blocking communication	-	
Argument	-			
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. * 1 The EvError event occurs only when the number of errors has decreased after the error reset is performed. Some errors are not cleared (system error, etc.).			
Detailed example C#	<pre>try{ axMBActX1.Operation.ClearError(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>MBActX1.Operation.ClearError</pre>			

SetCurrentJobNo

Sets the program number to be run on the unit.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvExpansionStart/End	Blocking communication	✓	
Argument	int	JobNo	Current program number	
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. At the time of transfer to the unit, if the program number is out of range The message, "Job No. range error", occurs.			
Detailed example C#	<pre>int JobNo = 10; try{ axMBActX1.Operation.SetCurrentJobNo(JobNo); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	See "GetCurrentJobNo".			

BeginTransaction

Starts changing the mark setting of the controller.

Type	Method	Mode	Operation	✓
Memory allocation	-	Blocking communication	Editing	✓
Event	-		-	
Argument	int	JobNo	Setting number	
Return value	-			
Note	When the mark setting saved in the controller is changed by using SetMethod of MBOperation in the order of "BeginTransaction" -> "Multiple executions of SetMethod in MBOperation" -> "CommitTransaction", expansion is performed only once at the time of the execution of CommitTransaction. This reduces the communication time. Executing CommitTransaction confirms the value, and executing RollbackTransaction resets the value to what it was when BeginTransaction was executed. Also, after BeginTransaction is executed and the setting data that had been changed before CommitTransaction or BeginTransaction was executed is referred to by a method such as Get**, the obtained value is handled as the changed value. After executing BeginTransaction, you must execute CommitTransaction/RollbackTransaction. If CommitTransaction/RollbackTransaction is not executed or the job exits without this method executed, the priority is taken from the unit and this status cannot be recovered until the controller is turned off and then turned on again. Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the program number is out of input range The out of input range error occurs.			
Detailed example C#	<pre>int JobNo = 10; int Column = 2; bool IsEnable = true; //Start of editing. try{ axMBActX1.Operation.BeginTransaction(JobNo); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); return; } //Multiple editing try{ for(int Row=1; Row < 10; Row++){ axMBActX1.Operation.SetMatrixCellMarkingEnable(JobNo, Row, Column, IsEnable); //Multiple editings } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); try{ axMBActX1.Operation.RollbackTransaction(); } //Discard the changes after BeginTransaction }catch{ MessageBox.Show(error.Message); } return; } //End of editing. try{ axMBActX1.Operation.CommitTransaction(); //Incorporate the changes after BeginTransaction }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); } }</pre>			
Simple example VBA	<pre>MBActX1.Operation.BeginTransaction 0 MBActX1.Operation.SetCharString2 0, 0, "111" MBActX1.Operation.SetCharString2 0, 1, "222" MBActX1.Operation.SetCharString2 0, 2, "333" If MBActX1.Operation.IsWarning = True Then MBActX1.Operation.RollbackTransaction End If MBActX1.Operation.CommitTransaction</pre>			

CommitTransaction

Finishes changing the mark setting of the controller.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvExpansion Start/End	Blocking communication	✓	
Argument	-			
Return value	-			
Note	Execute this method after executing BeginTransaction. Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Example	See BeginTransaction.			

RollbackTransaction

Cancels changing the mark setting of the controller.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	-			
Note	Execute this method after executing BeginTransaction. Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Example	See BeginTransaction.			

StartCameraScanningMode2

Starts camera scanning mode. The open/close status of the shutter can be specified in the finder mode.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	ShutterStateTypes	ShutterStateType	SHUTTERSTATE_OPEN //Open shutter	
			SHUTTERSTATE_CLOSE //Close shutter	
Return value	-			
Note	Specifies the open/close status of the shutter when the finder mode starts. The finder mode must be finished by FinishCameraScanningModeMethod. Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	try{ axMBActX1.Operation.StartCameraScanningMode2(ShutterStateTypes.SHUTTERSTATE_OPEN); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			
Simple example VBA	MBActX1.Operation.StartCameraScanningMode2 SHUTTERSTATE_CLOSE MBActX1.Operation.FinishCameraScanningMode			

FinishCameraScanningMode

Finishes camera scanning mode.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	-			
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>try{ axMBActX1.Operation.FinishCameraScanningMode(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	See "MBActX1.Operation.StartCameraScanningMode2"			

SetCameraPosition

Sets the camera position.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	double	PositionX//X coordinate (The thousandth place digit)		
	double	PositionY//Y coordinate (The thousandth place digit)		
	double	PositionZ//Z coordinate (The hundredth place digit)		
	bool	IsAffectWorkpiecePositionAdjustment	True : Reflect the workpiece position adjustment False : Do not reflect the workpiece position adjustment	
Return value	-			
Note	Make sure to execute StartCameraScanningMode2 method before entering the finder mode. To adjust the workpiece position by the image processing of the camera, set IsAffectWorkpiecePositionAdjustment to false because scanning is necessary at the coordinate where workpiece position is not adjusted. To change the block position by the image processing of the camera, set IsAffectWorkpiecePositionAdjustment to true because scanning is necessary at the coordinate where workpiece position has been adjusted. Exception: If the equipment is not connected with the unit The "Communication error" occurs. Before executing StartCameraScanningModeMethod An "unsupported error" occurs. If X, Y, or Z coordinate is out of input range The out of input range error occurs.			
Detailed example C#	<pre>double PositionX = 1.000; double PositionY = 2.000; double PositionZ = 3.00; bool IsAffectWorkpiecePositionAdjustment = false; try{ axMBAcX1.Operation.SetCameraPosition(PositionX, PositionY, PositionZ, IsAffectWorkpiecePositionAdjustment); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>MBAcX1.Operation.StartCameraScanningMode2 SHUTTERSTATE_CLOSE MBAcX1.Operation.SetCameraPosition 50, 50, 0, False MBAcX1.Operation.FinishCameraScanningMode</pre>			

OpenShutter

Opens the shutter.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvError	Blocking communication	-	
Argument	-			
Return value	-			
Note	The terminal block status of trigger being inhibited is released, and thus EvErrorEvent occurs. Use the IsShutterOpen property to check the open/close status of the shutter. Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>try{ axMBAcX1.Operation.OpenShutter(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBAcX1.Operation.OpenShutter			

CloseShutter

Closes the shutter.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvError	Blocking communication	-	
Argument	-			
Return value	-			
Note	Executing this method during marking cancels the marking and closes the shutter. Use the IsShutterOpen property to check the open/close status of the shutter. Trigger becomes inhibited in the terminal block, and thus EvErrorEvent occurs. Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>try{ axMBAcX1.Operation.CloseShutter(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBAcX1.Operation.CloseShutter			

SetMatrixCellOffset

Sets the offset of a matrix.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvExpansionStart/ End *1	Blocking communication	✓	
Argument	int	JobNo//Setting number		
	int	Reserved//Reserved (Fixed to 0)		
	int	Row//Row		
	int	Column//Column		
	double	OffsetX //Offset X coordinate (mm, the thousandth place digit)		
	double	OffsetY //Offset Y coordinate (mm, the thousandth place digit)		
	double	OffsetZ //Offset Z coordinate (mm, the thousandth place digit)		
	double	OffsetAngle//Offset angle (mm, the thousandth place digit)		
	double	MarkingDelayTime// Wait time for start cell marking (ms, the tenth place digit)		
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the program number is out of input range The out of input range error occurs. If the number of rows/columns is out of input range The out of input range error occurs. If the offset value is out of input range The out of input range error occurs. * 1 If the BeginTransaction method is used, the EvExpansionStart/End event does not occur.			
Detailed example C#	int JobNo = 5; int Row = 2; int Column = 3; double OffsetX = /2500; double OffsetY = -2.000; double OffsetZ = 0.500; double OffsetAngle = 3.000; double MarkingDelayTime = 10.0; try{ axMBActX1.Operation.GetMatrixCellOffset (JobNo, 0, Row, Column, OffsetX, OffsetY, OffsetZ, OffsetAngle, MarkingDelayTime); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); } }			
Simple example VBA	Dim OffsetX As Double Dim OffsetY As Double Dim OffsetZ As Double Dim OffsetAngle As Double Dim MDT As Double MBActX1.Operation.SetMatrixCellOffset 1, 0, 1, 1, 5, 5, 0, 45, 10 MBActX1.Operation.GetMatrixCellOffset 1, 0, 1, 1, OffsetX, OffsetY, OffsetZ, OffsetAngle, MDT Debug.Print OffsetX Debug.Print OffsetY Debug.Print OffsetZ Debug.Print OffsetAngle Debug.Print MDT			

GetMatrixCellOffset

Gets the offset of a matrix.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	JobNo//Setting number		
	int	Reserved//Reserved (Fixed to 0)		
	int	Row//Row		
	int	Column//Column		
	out double	OffsetX //Offset X coordinate (mm, the thousandth place digit)		
	out double	OffsetY //Offset Y coordinate (mm, the thousandth place digit)		
	out double	OffsetZ //Offset Z coordinate (mm, the thousandth place digit)		
	out double	OffsetAngle//Offset angle (mm, the thousandth place digit)		
	out double	MarkingDelayTime //Wait time for start cell marking (ms, the tenth place digit)		
Return value	-			
Note	Exception: If the equipment is not connected with the unit The “Communication error” occurs. If the program number is out of input range The out of input range error occurs. If the number of rows/columns is out of input range The out of input range error occurs.			
Detailed example C#	int JobNo = 5; int Row = 2; int Column = 3; double OffsetX; double OffsetY; double OffsetZ; double OffsetAngle; double MarkingDelayTime; try{ axMBActX1.Operation.GetMatrixCellOffset (JobNo, 0, Row, Column, out OffsetX, out OffsetY, out OffsetZ, out OffsetAngle, out MarkingDelayTime); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			
Simple example VBA	See “SetMatrixCellOffset”.			

SetMatrixCellMarkingEnable

Sets the marking flag of the matrix.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	EvExpansionStart/End *1	Blocking communication	✓	
Argument	int	JobNo //Setting number		
	int	Reserved //Reserved (Fixed to 0)		
	int	Row //Row		
	int	Column //Column		
	bool	IsEnable //Marking flag		
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the program number is out of input range The out of input range error occurs. If the number of rows/columns is out of input range The out of input range error occurs. If no matrix exists An "unsupported error" occurs. * 1 If the BeginTransaction method is used, the EvExpansionStart/End event does not occur.			
Detailed example C#	int JobNo = 5; int Row = 2; int Column = 3; bool IsEnable = true; try{ axMBActX1.Operation.SetMatrixCellMarkingEnable(JobNo, 0, Row, Column, IsEnable); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			
Simple example VBA	MBActX1.Operation.SetMatrixCellMarkingEnable 1, 0, 2, 2, False Debug.Print MBActX1.Operation.GetMatrixCellMarkingEnable(1, 0, 2, 2)			

GetMatrixCellMarkingEnable

Gets the marking flag of the matrix.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	JobNo //Setting number		
	int	Reserved //Reserved (Fixed to 0)		
	int	Row //Row		
	int	Column //Column		
Return value	bool	Marking flag		
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the program number is out of input range The out of input range error occurs. If the number of rows/columns is out of input range The out of input range error occurs. If no matrix exists An "unsupported error" occurs.			
Detailed example C#	<pre>int JobNo = 5; int Row = 2; int Column = 3; try{ bool IsEnable = axMBActX1.Operation.GetMatrixCellMarkingEnable(JobNo, 0, Row, Column); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	See "SetMatrixCellMarkingEnable".			

Read2DCode2

Reads a 2D code.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	Ev2DCodeReading Start/End	Blocking communication	-	
Argument	-			
	int	DecodingArea	Reading area (%: 0 to 100)	
Return value	bool	True: 2D code reading is successful. False: 2D code reading failed.		
Note	Regardless of the status, true/false, the IsBlockingCommunication function does not exit until the reading is completed. To scan a 2D code, the position of the camera finder must be adjusted in advance by using SetCameraPositionMethod. When the reading is successfully completed, use Get2DCodeReadResultMethod to get the reading result. Specify the proportion of the area to be used for reading to the whole imaging area, using DecodingArea. If the reading area is small, the reading time will be shortened, however, the reading error will occur when the area is too small for the code to be accommodated within the area. When DecodingArea is 0, reading is performed for the entire imaging area. Exception: If the equipment is not connected with the unit The "Communication error" occurs. For an unsupported model The message, "S095: Model limitation error", occurs. When the 2D code reader function is disabled The message, "S084: Operation limitation error", occurs.			
Detailed example C#	<pre>double DecodingArea = 70; string Content; int Grade; try{ if(axMBActX1.Operation.Read2DCode2(DecodingArea)){ axMBActX1.Operation.Get2DCodeReadResult(out Content, out Grade); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>MBActX1.Operation.Read2DCode2 70 MBActX1.Operation.Get2DCodeReadResult Content, grade If grade = 0 Then Debug.Print "F" Elseif grade = 1 Then Debug.Print "D" Elseif grade = 2 Then Debug.Print "C" Elseif grade = 3 Then Debug.Print "B" Elseif grade = 4 Then Debug.Print "A" Elseif grade = -1 Then Debug.Print "Reading Failed" End If</pre>			

Read2DCodePosition2

Reads a 2D code in a specified position.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	Ev2DCodeReadingStart/End	Blocking communication	-	
Argument	double	X //X coordinate of the 2D code to read (mm, the thousandth place digit)		
	double	Y //Y coordinate of the 2D code to read (mm, the thousandth place digit)		
	double	Z //Z coordinate of the 2D code to read (mm, the thousandth place digit)		
	int	DecodingArea //Reading area (%: 0 to 100)		
Return value	bool	True: 2D code reading is successful. False: 2D code reading failed.		
Note	Regardless of the status, true/false, the IsBlockingCommunication function does not exit until the reading is completed. When the reading is successfully completed, use Get2DCodeReadResultMethod to get the reading result. See Read2DCode2 for DecodingArea.			
Detailed example C#	<pre>double X = 10.000; double Y = -5.000; double Z = 1.200; int DecodingArea = 70; intstring Content; int Grade; try{ if(axMBActX1.Operation.Read2DCodePosition2(X, Y, Z, DecodingArea)){ axMBActX1.Operation.Get2DCodeReadResult(out Content, out Grade); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Operation.Read2DCodePosition2 10, 10, 0, 70			

Read2DCodeBlock

Reads a 2D code specified by block number and area.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	Ev2DCodeReadingStart/End	Blocking communication	-	
Argument	int	BlockNo //Block number of the 2D code to read		
	int	DecodingArea //Reading area (%: 0 to 100)		
Return value	bool	True: 2D code reading is successful. False: 2D code reading failed.		
Note	Regardless of the status, true/false, the IsBlockingCommunication function does not exit until the reading is completed. Use Get2DCodeReadResult to get the result after reading. Specify the proportion of the area to be used for reading to the whole imaging area, using DecodingArea. If the reading area is small, the reading time will be shortened, however, the reading error will occur when the area is too small for the code to be accommodated within the area. When DecodingArea is 0, reading is performed for the entire area.			
Detailed example C#	<pre>int BlockNo = 0; int DecodingArea = 0; intstring Content; int Grade; try{ if(axMBActX1.Operation.Read2DCodeBlock(BlockNo, DecodingArea)){ axMBActX1.Operation.Get2DCodeReadResult(out Content, out Grade); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Operation.Read2DCodeBlock 0, 0			

Get2DCodeReadResult

Gets the reading result of the last 2D code.

Type	Method		Mode	Operation	✓
Memory allocation	-			Editing	✓
Event	-		Blocking communication	-	
Argument	out string	Content	Reading contents (Decoding result)		
	out int	Grade	AIMDPM overall judgment grade (4:A, 3:B, 2:C, 1:D, 0:F)		
Return value	-				
Note	The 2D code must be read in advance by using the Read2DCode or Read2DCodePosition method. Exception: If the equipment is not connected with the unit The "Communication error" occurs. For an unsupported model The message, "S095: Model limitation error", occurs. When the 2D code reader function is disabled The message, "S084: Operation limitation error", occurs. If reading of the last 2D code failed An "unsupported error" occurs.				
Example	See Read2DCode.				

GetCameraViewImage

Gets the displayed image on the finder view in Bitmap format.

Type	Method		Mode	Operation	✓
Memory allocation	-			Editing	✓
Event	-		Blocking communication	-	
Argument	string	FilePath	File path of the image to save		
Return value	bool		True: Successfully saved False: Saving failed		
Note	The finder image to save must be displayed in advance by setting the IsCameraFinderView property to true. When only the file name is specified, the file is saved in the My Picture folder. However, if the FilePath argument is a null character (""), the "Save As" dialog starts where you can specify a file path. When a relative path is specified, the target file is saved in the relative path from the My Picture folder. Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the display mode of the preview is not Finder An "unsupported error" occurs.				
Detailed example C#	<pre>try{ if(axMBActX1.Operation.IsCameraFinderView){ axMBActX1.Operation. GetCameraViewImage("CameraImage.BMP"); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>				
Simple example VBA	<pre>MBActX1.Operation.IsCameraFinderView = True MBActX1.Operation.GetCameraViewImage ("test.BMP") MBActX1.Operation.IsCameraFinderView = False</pre>				

StartZTrackingCalibration

Performs the calibration of the Z tracking.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	✓	
Argument	-	-	-	
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the other device acquires mastership "Priority error" occurs.			
Detailed example C#	<pre>try{ axMBActX1.Operation.StartZTrackingCalibration(); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Call MBActX1.Operation.StartZTrackingCalibration			

CheckZPosition

Gets the Z coordinate of a specified point in the X/Y plane.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	✓	
Argument	double	x	X coordinate	
	double	y	Y coordinate	
	out double	stability	Stability	
Return value	double	Z coordinate of the specified X/Y coordinates		
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the other device acquires mastership "Priority error" occurs.			
Detailed example C#	<pre>try{ axMBActX1.Operation.CheckZPosition(x, y, out stability); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Call MBActX1.Operation.CheckZPosition(x, y, stability)			

StartZTracking

Executes the Z tracking.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	✓	
Argument	int	trackingNo	Z tracking No. to be executed (0 to 255)	
Return value	-			
Note	Exception: If the equipment is not connected with the unit The “Communication error” occurs. If the other device acquires mastership “Priority error” occurs.			
Detailed example C#	axMBAcTX1.Operation.StartZTracking(0);			
Simple example VBA	Call MBAcTX1.Operation.StartZTrackingCalibration			

GetZTrackingResult

Gets the correction result of the Z tracking.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	trackingNo	Z tracking No. to get results (0 to 255)	
	out double	diffZ	Z correction amount (mm)	
	out double	tiltX	Tilt correction amount X (mm)	
	out double	tiltY	Tilt correction amount Y (mm)	
	out int	stab	Stability	
	out int	time	Processing time (s)	
	out bool	isWithinRange	Within or out of measurement range	
	out bool	result	Execution result	
Return value	-			
Note	Exception: If the equipment is not connected with the unit The “Communication error” occurs.			
Detailed example C#	double diffZ; double tiltX; double tiltY; int stab; int time; bool isWithinTolerance; bool result; axMBActX1.Operation.GetZTrackingResult(0, out diffZ, out tiltX, out tiltY, out stab, out time, out isWithinRange, out result);			
Simple example VBA	Dim diffZ As Double Dim tiltX As Double Dim tiltY As Double Dim stab As Integer Dim time As Integer Dim isWithinTolerance As Boolean Dim result As Boolean Call MBActX1.Operation.GetZTrackingResult(0, diffZ, tiltX, tiltY, stab, time, isWithinTolerance, result)			

StartXYTracking

Executes the XY tracking.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	✓	
Argument	int	trackingNo	XY tracking No. to be executed (0 to 7)	
Return value	-			
Note	Exception: If the equipment is not connected with the unit The “Communication error” occurs.			
Detailed example C#	axMBActX1.Operation.StartXYTracking(0);			
Simple example VBA	Call MBActX1.Operation.StartXYTracking(0)			

GetXYTrackingResult

Gets the correction result of the XY tracking.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	int	trackingNo	XY tracking No. to get results (0 to 7)	
	out double	diffX	X correction amount (mm)	
	out double	diffY	Y correction amount (mm)	
	out double	angle	Correction angle (°)	
	out int	corr	Correlation value	
	out int	time	Processing time (s)	
	out bool	result	Execution result	
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	double diffX; double diffY; double angle; int corr; int time; bool result; axMBActX1.Operation.GetXYTrackingResult(0, out diffX, out diffY, out angle, out corr, out time, out result);			
Simple example VBA	Dim diffX As Double Dim diffY As Double Dim angle As Double Dim corr As Integer Dim time As Integer Dim result As Boolean Call MBActX1.Operation.GetXYTrackingResult(0, diffX, diffY, angle, corr, time, result)			

GetWindowCheckResult

Gets the result of the lens inspection before or after marking.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	WindowCheckTimingTypes	TimingType	WINDOW_CHECK_BEFORE_MARKING: Before marking WINDOW_CHECK_AFTER_MARKING: After marking	
	out int	score	Maintenance result	
	out int	time	Processing time (ms)	
Return value	-			
Note	Exception: If the equipment is not connected with the unit The “Communication error” occurs.			
Detailed example C#	int score; int time; axMBActX1.Operation.GetWindowCheckResult(WindowCheckTimingTypes.WINDOW_CHECK_BEFORE_MARKING, out score, out time);			
Simple example VBA	Dim score As Integer Dim time As Integer Call MBActX1.Operation.GetWindowCheckResult(WINDOW_CHECK_BEFORE_MARKING, score, time)			

StartWindowCheck

Executes the lens inspection before or after marking.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	✓	
Argument	WindowCheck TimingTypes	TimingType	WINDOW_CHECK_BEFORE_MARKING: Before marking WINDOW_CHECK_AFTER_MARKING: After marking	
Return value	int	-	Maintenance result	
Note	Exception: If the equipment is not connected with the unit The “Communication error” occurs. If the other device acquires mastership “Priority error” occurs.			
Detailed example C#	int score = axMBAcTX1.Operation.StartWindowCheck(WindowCheckTimingTypes.WINDOW_CHECK_BEFORE_MARKING);			
Simple example VBA	Dim score As Integer score = axMBAcTX1.Operation.StartWindowCheck(WINDOW_CHECK_BEFORE_MARKING)			

SetLighting

Sets the ON/OFF status of the lighting set by a job during operation.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	✓	
Argument	bool	lightingOn	True: lighting ON False: lighting OFF	
Return value	-			
Note	Exception: If the equipment is not connected with the unit The “Communication error” occurs. If the other device acquires mastership “Priority error” occurs.			
Detailed example C#	(light on) axMBAcTX1.Operation.SetLighting(true); (light off) axMBAcTX1.Operation.SetLighting(false);			
Simple example VBA	(light on) Call MBAcTX1.Operation.SetLighting(True) (light off) Call MBAcTX1.Operation.SetLighting(False)			

GetLighting

Gets the ON/OFF status of the lighting set by a job during operation.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Argument	bool	lightingOn	True: lighting ON False: lighting OFF	
Return value	-			
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the other device acquires mastership "Priority error" occurs.			
Detailed example C#	bool lightingOn = axMBAcTX1.Operation.GetLighting();			
Simple example VBA	Dim lightingOn As Boolean lightingOn = MBAcTX1.Operation.GetLighting			

Property

IsCameraScanningMode

Gets whether the camera is being controlled.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	bool	True: Controlling camera False: Not controlling camera		
Note	When polling this property in a way that the status is always displayed in the program, access the property at the interval of one second or more so that no communication load is applied. Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>try{ if (axMBActX1.Operation.IsCameraScanningMode){ MessageBox.Show("Camera Scanning Mode Now"); } else { MessageBox.Show("Normal Now"); } } catch (System.Runtime.InteropServices.COMException error) { MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Operation.IsCameraScanningMode			

IsExpanding

Gets whether expansion is in progress.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	bool	True: Expanding False: Not expanding		
Note	When polling this property in a way that the status is always displayed in the program, access the property at the interval of one second or more so that no communication load is applied. Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>try{ if (axMBActX1.Operation.IsExpanding){ MessageBox.Show("Expanding Now"); } else { MessageBox.Show("Not Expanding Now"); } } catch (System.Runtime.InteropServices.COMException error) { MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Operation.IsExpanding			

IsError

Gets whether an error has occurred.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	bool	True: Error has occurred. False: No error has occurred.		
Note	This property is synchronized with the error output of the controller terminal block output. When polling this property in a way that the status is always displayed in the program, access the property at the interval of one second or more so that no communication load is applied. Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>try{ if (axMBActX1.Operation.IsError){ MessageBox.Show("Error Status Now"); } else { MessageBox.Show("No Error Now"); } } catch (System.Runtime.InteropServices.COMException error) { MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Operation.IsError			

IsWarning

Gets whether a warning is being issued.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	bool	True: Warning being issued False: No warning being issued		
Note	This property is synchronized with the warning output of the controller terminal block output. When polling this property in a way that the status is always displayed in the program, access the property at the interval of one second or more so that no communication load is applied. Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>try{ if (axMBActX1.Operation.IsWarning){ MessageBox.Show("Warning Status Now"); } else { MessageBox.Show("No Warning Now"); } } catch (System.Runtime.InteropServices.COMException error) { MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Operation.IsWarning			

IsReady

Gets the Ready status.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	bool	True: READY False: Not READY		
Note	This property is synchronized with the READY output of the controller terminal block output. When polling this property in a way that the status is always displayed in the program, access the property at the interval of one second or more so that no communication load is applied. Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>try{ if (axMBActX1.Operation.IsReady){ MessageBox.Show("Ready Now"); } else { MessageBox.Show("Not Ready Now"); } } catch (System.Runtime.InteropServices.COMException error) { MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Operation.IsReady			

IsShutterOpen

Gets the open/close status of the shutter.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	bool	True: Shutter is open. False: Shutter is closed.		
Note	This property is synchronized with the shutter status output of the controller terminal block output. When polling this property in a way that the status is always displayed in the program, access the property at the interval of one second or more so that no communication load is applied. Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>try{ if (axMBActX1.Operation.IsShutterOpen){ MessageBox.Show("Shutter Open Now"); } else { MessageBox.Show("Shutter Closed Now"); } } catch (System.Runtime.InteropServices.COMException error) { MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Operation.IsShutterOpen			

IsMarking

Gets whether marking is in progress.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	bool	True: Marking False: Not marking		
Note	This property is synchronized with the marking output of the controller terminal block output. When polling this property in a way that the status is always displayed in the program, access the property at the interval of one second or more so that no communication load is applied. Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>try{ if (axMBActX1.Operation.IsMarking){ MessageBox.Show("Marking Now"); } else { MessageBox.Show("Not Marking Now"); } } catch (System.Runtime.InteropServices.COMException error) { MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Operation.IsMarking			

IsLaserActivated

Gets the laser indicator (whether the laser excitation is supplied).

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	bool	True: Excited laser state False: Unexcited laser state		
Note	This property is synchronized with the laser indicator output of the controller terminal block output. When polling this property in a way that the status is always displayed in the program, access the property at the interval of one second or more so that no communication load is applied. Exception: If the equipment is not connected with the unit The "Communication error" occurs.			
Detailed example C#	<pre>try{ if (axMBActX1.Operation.IsLaserActivated){ MessageBox.Show("Laser Activated Now"); } else { MessageBox.Show("Laser Deactivate Now"); } } catch (System.Runtime.InteropServices.COMException error) { MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Operation.IsLaserActivated			

IoEncodedCharacter

Gets/sets the I/O encoded value (Old term: Rank).

Type	Property (RW)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	I/O encoded value (0 to 35)		
Note	Exception: If the equipment is not connected with the unit The "Communication error" occurs. If the I/O encoded value is out of input range The out of input range error occurs.			
Detailed example C#	<pre>int CurrentRankValue = 5; try{ axMBActX1.Operation.IoEncodedCharacter(CurrentRankValue); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Operation.IoEncodedCharacter = 5 Debug.Print MBActX1.Operation.IoEncodedCharacter			

IsCameraFinderView

Gets/sets whether to set the preview display mode to Finder.

Type	Property (RW)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	bool	True: Camera mode False: Normal mode		
Note	The finder view is available only when the connection type is Ethernet. Exception: If the equipment is not connected with the unit The "Communication error" occurs. For USB connection An "unsupported error" occurs.			
Detailed example C#	<pre>try{ if(axMBActX1.Operation.IsCameraFinderView){ axMBActX1.Operation.GetCameraViewImage("CameraImage.BMP"); } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Operation.IsCameraFinderView = True Debug.Print MBActX1.Operation.IsCameraFinderView			

MBError

Property

ErrorCode

Gets the error code number.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	String	Error number		
Note	Before using ErrorCode, UpdateErrorCount needs to be executed. If UpdateErrorCount is executed it saves the information from prior errors in the cache. ErrorCode (and ErrorCaption of MBError class) gets the information from the cache. Exception: When specifying a number with "Error(number).ErrorCode", specify a number where the error does not exist Out of range error An example when an "Out of range error" occurs : When there is only one error occurred, before executing "UpdateErrorCount", if "Error(0).ErrorCode" is executed, an "Out of range error" occurs. (Before executing "UpdateErrorCount", the cache is empty). After executing "UpdateErrorCount", if "Error(1).ErrorCode" is executed, an "Out of range error" occurs. (since there is only one error, "Error(0).ErrorCode" is correct)			
Detailed example C#	<pre>int Count; string List = ""; try{ Count = axMBActX1.UpdateErrorCount(); for (int index = 0; index < Count; index++){ List = List + axMBActX1.Error(index).ErrorCode + ":" + axMBActX1.Error(index).ErrorCaption + "\n"; } MessageBox.Show("Occurring error list is below ... \n" + List); } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.UpdateErrorCount MsgBox MBActX1.Error(0).ErrorCode & " " & MBActX1.Error(0).ErrorCaption			

ErrorCaption

Gets the error description.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	string	Error description		
Note	Before using ErrorCaption, UpdateErrorCount needs to be executed. See the note of ErrorCode.			
Example	See ErrorCode.			

MBErrorHistory

Property

ErrorCode

Gets the error codes registered in the error history.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	string	Error number		

Note

Before using ErrorCode, UpdateErrorHistoryCount needs to be executed.
If UpdateErrorHistoryCount is executed it will save the error history in the cache. ErrorCode (and ErrorCaption and ErrorDateTime of MBErrorHistory class) gets the information from the cache.

Exception:

When specifying a number with "ErrorHistory(number).ErrorCode", specify a number that does not exist in the error history.
Out of range error

Detailed example C#

```
int Count;
string List = "";

try{
    Count = axMBActX1.UpdateErrorHistoryCount();
    for (int index = 0; index < Count; index++){
        List = List + "[" + axMBActX1.ErrorHistory(index).ErrorDateTime.ToString() + "]"
        + axMBActX1.ErrorHistory(index).ErrorCode + " " + axMBActX1.ErrorHistory(index).ErrorCaption + "\n";
    }
    MessageBox.Show("Error history list is below ... \n" + List);
}
catch (System.Runtime.InteropServices.COMException error){
    MessageBox.Show(error.Message);
}
```

Simple example VBA

MBActX1.UpdateErrorHistoryCount
MsgBox MBActX1.ErrorHistory(0).ErrorCode & " " & MBActX1.ErrorHistory(0).ErrorCaption & " " & MBActX1.ErrorHistory(0).ErrorDateTime

ErrorCaption

Gets the error information of the errors registered in the error history.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	string	Error description		
Note	Before using ErrorCaption, UpdateErrorHistoryCount needs to be executed. See the note of ErrorCode.			
Example	See ErrorCode.			

ErrorDateTime

Gets the occurrence date and time of the errors registered in the error history.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	DateTime	Error occurrence date and time		
Note	Before using ErrorCaption, UpdateErrorHistoryCount needs to be executed. See the note of ErrorCode.			
Example	See ErrorCode.			

MBControllerFile

Property

FileName

Gets the file name.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	string	File name		
Note	Before using FileName, UpdateControllerFileCount needs to be executed. If UpdateControllerFileCount is executed, save the file name information in the cache. According to the argument of UpdateControllerFileCount, the file type to be cached changes. • CONTROLLERFILE_CHARACTER: Custom character • CONTROLLERFILE_FONT: System font • CONTROLLERFILE_LOGO: Logo • CONTROLLERFILE_JOB: Job • CONTROLLERFILE_ZMAP: ZMAP FileName gets the information from the cache. Exception: When specifying a number with ControllerFile(number).FileName, specify a number where the file name does not exist Out of range error			
Detailed example C#	<pre>int Count; string List = ""; try{ Count = axMBActX1.UpdateErrorCount(); for (int index = 0; index < Count; index++){ List = List + axMBActX1.Error(index).ErrorCode + ":" + axMBActX1.Error(index).ErrorCaption + "\n"; } MessageBox.Show("Occurring error list is below ... \n" + List); } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	<pre>MBActX1.UpdateErrorCount MsgBox MBActX1.Error(0).ErrorCode & " " & MBActX1.Error(0).ErrorCaption</pre>			

JobNo

Gets the program No.

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	Int	Setting number		
Note	Before using JobNo, UpdateControllerFileCount (the argument is CONTROLLERFILE_JOB) needs to be executed. If UpdateControllerFileCount is executed, save the job number information in the cache. Jbo No gets the information from the cache. Exception: When specifying a number with ControllerFile(number).FileName, specify a number where the file name does not exist Out of range error			
Detailed example C#	int Count; string List = ""; try{ Count = axMBActX1.UpdateControllerFileCount(MBPLib1.ControllerFileTypes.CONTROLLERFILE_JOB); for (int index = 0; index < Count; index++){ List = List + axMBActX1.ControllerFile(index).JobNo + ":" + axMBActX1.ControllerFile(index).FileName + "\n"; } MessageBox.Show("controller Job file list is below ... \n" + List); } catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			
Simple example VBA	MBActX1.UpdateControllerFileCount CONTROLLERFILE_JOB MsgBox MBActX1.ControllerFile(0).JobNo & " " & MBActX1.ControllerFile(0).Filename			

MBDxf

Method

Convert

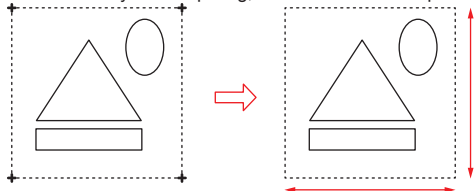
Convert the DXF file to a hatch logo file.

Type	Method	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	string	DxfFileName	DXF file name	
	string	OutputFileName	Hatch logo file name after conversion	
	bool	IsShowProgress	Whether to display the progress dialog	
Note	When an absolute path is specified for the file name, the target file is saved to the specified path. When the file name only is specified, open the file in the logo/photo data folder. (C:\Users\{User name}\Documents\Marking Builder Plus\Logo) Include the “mhl” extension in the hatch logo file name after conversion. If the same mhl file already exists, overwrite it. Exception: If no file exists The message, “No-File Error”, occurs. If no extension exists The message, “Failed to open transmission file”, occurs.			
Detailed example C#	string DxfFile = “test.dxf”; string LogoFile = “test.mhl”; bool ShowProgress = true; try{ axMBActX1.Dxf.Convert(DxfFile, LogoFile, ShowProgress); }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }			
Simple example VBA	MBActX1.dxf.Convert “test.dxf”, “test.mhl”, True			

Property

IsDetectPointElement

Sets whether to use the POINT element of the DXF file, as the DXF conversion condition, for computing the figure existing range of the file to output.

Type	Property (RW)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Return value	bool	True: Use the POINT element to compute the figure existing range False: Do not use the POINT element to compute the figure existing range		
Note	The default is true. Without the POINT element, this property is ignored. POINT is used only for computing, not written to the output file.  Convert the existing range including POINT as the logo size When [Use the POINT Element to Compute the Figure Existing Range] is set to true, a non-marked diagonal line from the lower left corner toward the upper right corner is added to the hatch logo file to be output.			
Detailed example C#	<pre>try{ axMBActX1.Dxf.IsDetectPointElement = true; }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.dxf.IsDetectPointElement = True Debug.Print MBActX1.dxf.IsDetectPointElement			

MBOverprintingCode

Property

CodeForm

Gets the code format (Normal/Black/White Inversion/Base).

Type	Property(R)	Mode	Operation	✓
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	CodeFormTypes	CODEFORM_NORMAL //Normal (return value : 0)		
		CODEFORM_REVERSE //Black/White Inversion (return value : 1)		
		CODEFORM_BASE //Base (return value : 2)		
Note	The return value is 0 to 2 in order from the top. Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; int index = 3; try{ if(axMBActX1.Block(blockNo).IsOverprintingCode){ CodeFormTypes cordform = axMBActX1.Block(blockNo).OverprintingCode(index).CodeForm; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	Debug.Print MBActX1.Block(0).OverprintingCode(0).CodeForm			

LaserPower

Gets/sets the laser power of a multi-pass code.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Laser power (% , The tenth place digit)		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; int index = 3; try{ if(axMBActX1.Block(blockNo).IsOverprintingCode && !axMBActX1.Block(blockNo).OverprintingCode(index).IsLaserPowerBlockCommon){ axMBActX1.Block(blockNo).OverprintingCode(index).LaserPower = 50.0; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Block(0).OverprintingCode(0).IsLaserPowerBlockCommon = False MBActX1.Block(0).OverprintingCode(0).LaserPower = 50 Debug.Print MBActX1.Block(0).OverprintingCode(0).LaserPower			

IsLaserPowerBlockCommon

Gets/sets whether to use the common laser power parameter for blocks.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	bool	TRUE : Common laser power parameter for blocks is enabled. FALSE : Laser power of overprinting is enabled.		
Note	-			
Example	See LaserPower.			

ScanSpeed

Gets/sets the scan speed of multi-pass code.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Scan speed (mm/s)		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; int index = 3; try{ if(axMBActX1.Block(blockNo).IsOverprintingCode && !axMBActX1.Block(blockNo).OverprintingCode(index).IsScanSpeedBlockCommon){ axMBActX1.Block(blockNo).OverprintingCode(index).ScanSpeed = 1230; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Block(0).OverprintingCode(0).IsScanSpeedBlockCommon = False MBActX1.Block(0).OverprintingCode(0).ScanSpeed = 1000 Debug.Print MBActX1.Block(0).OverprintingCode(0).ScanSpeed			

IsScanSpeedBlockCommon

Gets/sets whether to use the common scan speed parameter for blocks.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	bool	TRUE : Common scan speed parameter for blocks is enabled. FALSE : Scan speed of overprinting is enabled.		
Note	-			
Example	See ScanSpeed.			

Frequency

Gets/sets the pulse frequency of multi-pass code.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Pulse frequency (kHz)		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; int index = 3; try{ if(axMBActX1.Block(blockNo).IsOverprintingCode && !axMBActX1.Block(blockNo).OverprintingCode(index).IsFrequencyBlockCommon){ axMBActX1.Block(blockNo).OverprintingCode(index).Frequency = 150; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Block(0).OverprintingCode(0).IsFrequencyBlockCommon = False MBActX1.Block(0).OverprintingCode(0).Frequency = 30 Debug.Print MBActX1.Block(0).OverprintingCode(0).Frequency			

IsFrequencyBlockCommon

Gets/sets whether to use the common pulse frequency parameter for blocks.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-		Blocking communication	-
Type	bool	TRUE: Common pulse frequency parameter for blocks is enabled. FALSE: Pulse frequency of overprinting is enabled.		
Note	-			
Example	See Frequency.			

SpotVariable

Gets/sets the spot variable of multi-pass code.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Spot variable		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; int index = 3; try{ if(axMBActX1.Block(blockNo).IsOverprintingCode && !axMBActX1.Block(blockNo).OverprintingCode(index).IsSpotVariableBlockCommon){ axMBActX1.Block(blockNo).OverprintingCode(index).SpotVariable = -50; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Block(0).OverprintingCode(0).IsSpotVariableBlockCommon = False MBActX1.Block(0).OverprintingCode(0).SpotVariable = -20 Debug.Print MBActX1.Block(0).OverprintingCode(0).SpotVariable			

IsSpotVariableBlockCommon

Gets/sets whether to use the common spot variable parameter for blocks.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	bool	TRUE: Common spot variable parameter for blocks is enabled. FALSE: Spot variable of overprinting is enabled.		
Note	-			
Example	See SpotVariable.			

Repetition

Gets/sets the repetition of multi-pass code.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Repetition		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; int index = 3; try{ if(axMBActX1.Block(blockNo).IsOverprintingCode && !axMBActX1.Block(blockNo).OverprintingCode(index).IsRepetitionBlockCommon){ axMBActX1.Block(blockNo).OverprintingCode(index).Repetition = 3; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Block(0).OverprintingCode(0).IsRepetitionBlockCommon = False MBActX1.Block(0).OverprintingCode(0).Repetition = 3 Debug.Print MBActX1.Block(0).OverprintingCode(0).Repetition			

IsRepetitionBlockCommon

Gets/sets whether to use the common repetition parameter for blocks.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-		Blocking communication	-
Type	bool	TRUE: Common repetition parameter for blocks is enabled. FALSE: Repetition of overprinting is enabled.		
Note	-			
Example	See Repetition.			

FillLineInterval

Gets/sets the fill interval of multi-pass code.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Fill interval (μm)		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; int index = 3; try{ if(axMBActX1.Block(blockNo).IsOverprintingCode && !axMBActX1.Block(blockNo).OverprintingCode(index).IsFillLineIntervalBlockCommon){ axMBActX1.Block(blockNo).OverprintingCode(index).FillLineInterval = 40; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Block(0).OverprintingCode(0).IsFillLineIntervalBlockCommon = False MBActX1.Block(0).OverprintingCode(0).FillLineInterval = 40 Debug.Print MBActX1.Block(0).OverprintingCode(0).FillLineInterval			

IsFillLineIntervalBlockCommon

Gets/sets whether to use the common line interval parameter for blocks.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	bool	TRUE: Common line interval parameter for blocks is enabled. FALSE: Fill interval of overprinting is enabled.		
Note	-			
Example	See FillLineInterval.			

FillMargin

Gets/sets the shrink fill of a multi-pass code.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Shrink fill (μm)		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; int index = 3; try{ if(axMBActX1.Block(blockNo).IsOverprintingCode){ axMBActX1.Block(blockNo).OverprintingCode(index).FillMargin = 10; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Block(0).OverprintingCode(0).FillMargin = 10 Debug.Print MBActX1.Block(0).OverprintingCode(0).FillMargin			

PatternBarcode

Gets/sets the pattern of a multi-pass barcode.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	PatternBarcode Types	PATTERNBARCODE_ALTERNATE //Alternate (return value : 0)		
		PATTERNBARCODE_UNIDIRECTIONAL //Unidirectional (return value : 1)		
		PATTERNBARCODE_BASEALTERNATE //Base: Alternate (return value : 2)		
		PATTERNBARCODE_BASEUNIDIRECTIONAL //Base: Unidirectional (return value : 3)		
		PATTERNBARCODE_BASEHORZALTERNATE //Base: Alternate (Horiz.) (return value : 4)		
		PATTERNBARCODE_BASEHORZUNIDIRECTIONAL //Base: Unidirectional (Horiz.) (return value : 5)		
Note	The return value is 0 to 5 in order from the top. Only the overprinting of barcode blocks is valid. Exception: If out of input range The out of input range error occurs. For an unsupported block An "unsupported error" occurs.			
Detailed example C#	<pre>int blockNo = 0; int index = 3; try{ if(axMBActX1.Block(blockNo).IsOverprintingCode){ axMBActX1.Block(blockNo).OverprintingCode(index).PatternBarcode = PatternBarcodeTypes.PATTERNBARCODE_ALTERNATE; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Block(1).OverprintingCode(1).PatternBarcode = PATTERNBARCODE_ALTERNATE Debug.Print MBActX1.Block(1).OverprintingCode(1).PatternBarcode			

Pattern2D

Gets/sets the pattern of a multi-pass 2D code.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	Pattern2DTypes	PATTERN2D_SYSTEM	//Common (return value : 0)	
		PATTERN2D_1	//Pattern 1 (return value : 1)	
		PATTERN2D_2	//Pattern 2 (return value : 2)	
		PATTERN2D_3	//Pattern 3 (return value : 3)	
		PATTERN2D_4	//Pattern 4 (return value : 4)	
		PATTERN2D_5	//Pattern 5 (return value : 5)	
		PATTERN2D_6	//Pattern 6 (return value : 6)	
		PATTERN2D_7	//Pattern 7 (return value : 7)	
		PATTERN2D_A	//Pattern A (return value : 8)	
		PATTERN2D_B	//Pattern B (return value : 9)	
		PATTERN2D_C	//Pattern C (return value : 10)	
		PATTERN2D_D	//Pattern D (1return value : 11)	
		PATTERN2D_E	//Pattern E (return value : 12)	
		PATTERN2D_F	//Pattern F (return value : 13)	
		PATTERN2D_ENTIRE1	//Entire pattern 1 (return value : 14)	
		PATTERN2D_ENTIRE2	//Entire pattern 2 (return value : 15)	
		PATTERN2D_ENTIRE3	//Entire pattern 3 (return value : 16)	
		PATTERN2D_ENTIRE4	//Entire pattern 4 (return value : 17)	
		PATTERN2D_INDIVIDUAL	//Individual (return value : 18)	
		PATTERN2D_B1	//Base: B1 (return value : 19)	
		PATTERN2D_B2	//Base: B2 (return value : 20)	
		PATTERN2D_B3	//Base: B3 (return value : 21)	
		PATTERN2D_B4	//Base: B4 (return value : 22)	
		PATTERN2D_B5	//Base: B5 (return value : 23)	
		PATTERN2D_B6	//Base: B6 (return value : 24)	
		PATTERN2D_B7	//Base: B7 (return value : 25)	
		PATTERN2D_B8	//Base: B8 (return value : 26)	
		PATTERN2D_B9	//Base: B9 (return value : 27)	
		PATTERN2D_ENTIRE5	//Entire pattern 5 (return value : 28)	
		PATTERN2D_ENTIRE6	//Entire pattern 6 (return value : 29)	
Note	The return value is 0 to 29 in order from the top. Only overprinting of 2D code blocks is valid. Exception: If out of input range The out of input range error occurs. For an unsupported block An “unsupported error” occurs.			
Detailed example C#	<pre>int blockNo = 0; int index = 3; try{ if(axMBActX1.Block(blockNo).IsOverprintingCode){ axMBActX1.Block(blockNo).OverprintingCode(index).Pattern2D = Pattern2DTypes.PATTERN2D_ENTIRE4; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.Block(0).OverprintingCode(0).Pattern2D = PATTERN2D_ENTIRE4 Debug.Print MBActX1.Block(0).OverprintingCode(0).Pattern2D			

MB3DShape

Property

X

Gets/sets the X coordinate of a 3D shape.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	X coordinate (mm, the thousandth place digit)		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int shapeNo = 0; try{ if(axMBAcX1.Shape(shapeNo) != null){ axMBAcX1.Shape(shapeNo).X = 10.000; axMBAcX1.Shape(shapeNo).Y = 20.000; axMBAcX1.Shape(shapeNo).Z = -3.000; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBAcX1.Shape(0).X = 5 Debug.Print MBAcX1.Shape(0).X			

Z

Gets/sets the Z coordinate of a 3D shape.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Z coordinate (mm, the thousandth place digit)		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int shapeNo = 0; try{ if(axMBAcX1.Shape(shapeNo) != null){ axMBAcX1.Shape(shapeNo).X = 10.000; axMBAcX1.Shape(shapeNo).Y = 20.000; axMBAcX1.Shape(shapeNo).Z = -3.000; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBAcX1.Shape(0).Z = 5 Debug.Print MBAcX1.Shape(0).Z			

Y

Gets/sets the Y coordinate of a 3D shape.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Y coordinate (mm, the thousandth place digit)		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int shapeNo = 0; try{ if(axMBAcX1.Shape(shapeNo) != null){ axMBAcX1.Shape(shapeNo).X = 10.000; axMBAcX1.Shape(shapeNo).Y = 20.000; axMBAcX1.Shape(shapeNo).Z = -3.000; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBAcX1.Shape(0).Y = 5 Debug.Print MBAcX1.Shape(0).Y			

MBCommonBlock

Property

LaserPower

Gets/sets the common laser power parameter for blocks.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Laser power (%; The tenth place digit)		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo). IsLaserPowerBlockCommon) { axMBActX1.CommonBlock.LaserPower = 50.0; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.CommonBlock.LaserPower = 80 Debug.Print MBActX1.CommonBlock.LaserPower			

Frequency

Gets/sets the common pulse frequency parameter for blocks.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Pulse frequency (kHz)		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsFrequencyBlockCommon) { axMBActX1.CommonBlock.Frequency = 90; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.CommonBlock.Frequency = 20 Debug.Print MBActX1.CommonBlock.Frequency			

ScanSpeed

Gets/sets the common scan speed parameter for blocks.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Scan speed (mm/s)		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo). IsScanSpeedBlockCommon) { axMBActX1.CommonBlock.ScanSpeed = 1230; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.CommonBlock.ScanSpeed = 1000 Debug.Print MBActX1.CommonBlock.ScanSpeed			

SpotVariable

Gets/sets the common spot variable parameter for blocks.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Spot variable		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo). IsSpotVariableBlockCommon) { axMBActX1.CommonBlock.SpotVariable = -10; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.CommonBlock.SpotVariable = -20 Debug.Print MBActX1.CommonBlock.SpotVariable			

Repetition

Gets/sets the common repetition parameter for blocks.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Repetition		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsRepetitionBlockCommon) { axMBActX1.CommonBlock.Repetition = 2; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.CommonBlock.Repetition = 3 Debug.Print MBActX1.CommonBlock.Repetition			

QualityLevel

Gets/sets the common quality level parameter for blocks.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	QualityLevelTypes	QUALITYLEVEL_TOPSPEED //Top speed (return value : 10)		
		QUALITYLEVEL_HIGHSPEED //High speed (return value : 11)		
		QUALITYLEVEL_STANDARD //Standard (return value : 12)		
		QUALITYLEVEL_HIGHQUALITY //High quality (return value : 13)		
		QUALITYLEVEL_TOPQUALITY //Top quality (return value : 14)		
Note	The return value is 10 to 14 in sequence from the top. Refer to the following when the quality level is specified to barcode or 2D code, which is an item in overprinting settings. Select "Set" to overprint all. Select "Get" to get the conditions of overprinting order being 0. Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo). IsQualityLevelBlockCommon) { axMBActX1.CommonBlock.QualityLevel = QualityLevelTypes.QUALITYLEVEL_STANDARD; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.CommonBlock.QualityLevel = QUALITYLEVEL_TOPQUALITY Debug.Print MBActX1.CommonBlock.QualityLevel			

Z

Gets/sets the common Z coordinate parameter for blocks.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Z coordinate (mm, the thousandth place digit)		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsZBlockCommon) { axMBActX1.CommonBlock.Z = -1.000; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.CommonBlock.Z = 0 Debug.Print MBActX1.CommonBlock.Z			

FillLineInterval

Gets/sets the common fill interval parameter for blocks.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	Fill interval (μm)		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo). IsFillLineIntervalBlockCommon) { axMBActX1.CommonBlock.FillLineInterval = 50; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.CommonBlock.FillLineInterval = 60 Debug.Print MBActX1.CommonBlock.FillLineInterval			

SkipCross

Gets/sets the common skip cross parameter for blocks.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	double	Skip cross (mm, the thousandth place digit)		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	<pre>int blockNo = 0; try{ if(axMBActX1.Block(blockNo).IsSkipCrossBlockCommon) { axMBActX1.CommonBlock.SkipCross = 0.500; } }catch (System.Runtime.InteropServices.COMException error){ MessageBox.Show(error.Message); }</pre>			
Simple example VBA	MBActX1.CommonBlock.SkipCross = 0.2 Debug.Print MBActX1.CommonBlock.SkipCross			

MBWindowCheck

Property

Enabled

Gets/sets whether or not to enable the lens inspection before or after marking.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	bool	-		
Note	-			
Detailed example C#	MBWindowCheck windowCheck = axMBActX1.GetWindowCheck(WindowCheckTimingTypes.WINDOW_CHECK_BEFORE_MARKING); if (windowCheck.Enabled == false) { windowCheck.Enabled = true; }			
Simple example VBA	Dim windowCheck As MBWindowCheck windowCheck = MBActX1.GetWindowCheck(WINDOW_CHECK_BEFORE_MARKING) If windowCheck.Enabled = False Then windowCheck.Enabled = True End If			

AlarmThreshold

Gets/sets the warning threshold of the lens inspection before or after marking. (When the threshold is set to "0", the warning will be disabled.)

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	int	-		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	MBWindowCheck windowCheck = axMBActX1.GetWindowCheck(WindowCheckTimingTypes.WINDOW_CHECK_BEFORE_MARKING); windowCheck.AlarmThreshold = 20;			
Simple example VBA	Dim windowCheck As MBWindowCheck windowCheck = MBActX1.GetWindowCheck(WINDOW_CHECK_BEFORE_MARKING) windowCheck.AlarmThreshold = 20			

Sensitivity

Gets/sets the sensitivity adjustment of the lens inspection before or after marking.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-		Editing	✓
Event	-	Blocking communication	-	
Type	WindowCheck SensitivityTypes	Sensitivity adjustment WINDOW_CHECK_SENSITIVITY_BASIC: standard WINDOW_CHECK_SENSITIVITY_HIGH1: high sensitivity 1 WINDOW_CHECK_SENSITIVITY_HIGH2: high sensitivity 2		
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	MBWindowCheck windowCheck = axMBActX1.GetWindowCheck(WindowCheckTimingTypes.WINDOW_CHECK_BEFORE_MARKING); windowCheck.Sensitivity = WindowCheckSensitivityTypes.WINDOW_CHECK_SENSITIVITY_HIGH2;			
Simple example VBA	Dim windowCheck As MBWindowCheck windowCheck = MBActX1.GetWindowCheck(WINDOW_CHECK_BEFORE_MARKING) windowCheck.Sensitivity = WindowCheckSensitivityTypes.WINDOW_CHECK_SENSITIVITY_HIGH2			

MBXYTracking

Property

Enable

Gets/sets whether or not to enable the XY tracking.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-	Mode	Editing	✓
Event	-		-	-
Type	bool	Blocking communication	-	-
Note	-			
Detailed example C#	MBXYTracking xyTracking = axMBActX.GetXYTracking(0); xyTracking.Enable = true;			
Simple example VBA	Dim xyTracking As MBXYTracking xyTracking = MBActX1.GetXYTracking(0) xyTracking.Enable = True			

CorrectionThreshold

Gets/sets the correlation value lower limit of the XY tracking.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-	Mode	Editing	✓
Event	-		-	-
Type	int	Blocking communication	-	-
Note	When the correlation value of the XY tracking operation result falls below the set correlation value lower limit, an error or warning will occur. Exception: If out of input range The out of input range error occurs.			
Detailed example C#	MBXYTracking xyTracking = axMBActX.GetXYTracking(0); xyTracking.CorrectionThreshold = 20;			
Simple example VBA	Dim xyTracking As MBXYTracking xyTracking = MBActX1.GetXYTracking(0) xyTracking.CorrectionThreshold = 20			

MBZTracking

Property

Enable

Gets/sets whether or not to enable the Z tracking.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-	Mode	Editing	✓
Event	-		-	-
Type	bool	Blocking communication	-	-
Note	Enabled/Disabled			
Detailed example C#	MBZTracking zTracking = axMBActX.GetZTracking(0); zTracking.Enable = true;			
Simple example VBA	Dim zTracking As MBZTracking zTracking = MBActX1.GetZTracking(0) zTracking.Enable = True			

Method

GetHeightLimit

Gets the correction amount threshold settings of the Z tracking.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-	Mode	Editing	✓
Event	-		-	-
Type	out double	upperLimit	Upper limit (-21.0 to 42.0) (mm)	-
Type	out double	lowerLimit	Lower limit (-42.0 to 21.0) (mm)	-
Note	-			
Detailed example C#	MBZTracking zTracking = axMBActX.GetZTracking(0); double upper; double lower; zTracking.GetHeightLimit(out upper, out lower);			
Simple example VBA	Dim zTracking As MBZTracking zTracking = MBActX1.GetZTracking(0) Dim upperLimit As Double Dim lowerLimit As Double Call zTracking.GetTolerance(upperLimit, lowerLimit)			

SetHeightLimit

Sets the correction amount threshold settings of the Z tracking.

Type	Property (RW)	Mode	Operation	-
Memory allocation	-	Mode	Editing	✓
Event	-		-	-
Type	double	upperLimit	Upper limit (-21.0 to 42.0) (mm)	-
Type	double	lowerLimit	Lower limit (-42.0 to 21.0) (mm)	-
Note	Exception: If out of input range The out of input range error occurs.			
Detailed example C#	MBZTracking zTracking = axMBActX.GetZTracking(0); zTracking.SetHeightLimit(11.1, -21.0);			
Simple example VBA	Dim zTracking As MBZTracking zTracking = MBActX1.GetZTracking(0) Call zTracking.SetHeightLimit(11.1, -21)			

Revision history

Revision history	Edition number	Revision details
March 2020	Official release	
June 2020	2nd edition	

WARRANTIES AND DISCLAIMERS

- (1) KEYENCE warrants the Products to be free of defects in materials and workmanship for a period of one (1) year from the date of shipment. If any models or samples were shown to Buyer, such models or samples were used merely to illustrate the general type and quality of the Products and not to represent that the Products would necessarily conform to said models or samples. Any Products found to be defective must be shipped to KEYENCE with all shipping costs paid by Buyer or offered to KEYENCE for inspection and examination. Upon examination by KEYENCE, KEYENCE, at its sole option, will refund the purchase price of, or repair or replace at no charge any Products found to be defective. This warranty does not apply to any defects resulting from any action of Buyer, including but not limited to improper installation, improper interfacing, improper repair, unauthorized modification, misapplication and mishandling, such as exposure to excessive current, heat, coldness, moisture, vibration or outdoors air. Components which wear are not warranted.
- (2) KEYENCE is pleased to offer suggestions on the use of its various Products. They are only suggestions, and it is Buyer's responsibility to ascertain the fitness of the Products for Buyer's intended use. KEYENCE will not be responsible for any damages that may result from the use of the Products.
- (3) The Products and any samples ("Products/Samples") supplied to Buyer are not to be used internally in humans, for human transportation, as safety devices or fail-safe systems, unless their written specifications state otherwise. Should any Products/Samples be used in such a manner or misused in any way, KEYENCE assumes no responsibility, and additionally Buyer will indemnify KEYENCE and hold KEYENCE harmless from any liability or damage whatsoever arising out of any misuse of the Products/Samples.
- (4) OTHER THAN AS STATED HEREIN, THE PRODUCTS/SAMPLES ARE PROVIDED WITH NO OTHER WARRANTIES WHATSOEVER. ALL EXPRESS, IMPLIED, AND STATUTORY WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF PROPRIETARY RIGHTS, ARE EXPRESSLY DISCLAIMED. IN NO EVENT SHALL KEYENCE AND ITS AFFILIATED ENTITIES BE LIABLE TO ANY PERSON OR ENTITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, PUNITIVE, SPECIAL OR CONSEQUENTIAL DAMAGES (INCLUDING, WITHOUT LIMITATION, ANY DAMAGES RESULTING FROM LOSS OF USE, BUSINESS INTERRUPTION, LOSS OF INFORMATION, LOSS OR INACCURACY OF DATA, LOSS OF PROFITS, LOSS OF SAVINGS, THE COST OF PROCUREMENT OF SUBSTITUTED GOODS, SERVICES OR TECHNOLOGIES, OR FOR ANY MATTER ARISING OUT OF OR IN CONNECTION WITH THE USE OR INABILITY TO USE THE PRODUCTS, EVEN IF KEYENCE OR ONE OF ITS AFFILIATED ENTITIES WAS ADVISED OF A POSSIBLE THIRD PARTY'S CLAIM FOR DAMAGES OR ANY OTHER CLAIM AGAINST BUYER. In some jurisdictions, some of the foregoing warranty disclaimers or damage limitations may not apply.

BUYER'S TRANSFER OBLIGATIONS:

If the Products/Samples purchased by Buyer are to be resold or delivered to a third party, Buyer must provide such third party with a copy of this document, all specifications, manuals, catalogs, leaflets and written information provided to Buyer pertaining to the Products/Samples.

Specifications are subject to change without notice.

KEYENCE CORPORATION

1-3-14, Higashi-Nakajima, Higashi-Yodogawa-ku, Osaka, 533-8555, Japan PHONE: +81-6-6379-2211

www.keyence.com/glb

AUSTRIA

Phone: +43 2236 378266 0

BELGIUM

Phone: +32 15 281 222

BRAZIL

Phone: +55-11-3045-4011

CANADA

Phone: +1-905-366-7655

CHINA

Phone: +86-21-3357-1001

CZECH REPUBLIC

Phone: +420 220 184 700

FRANCE

Phone: +33-1-56-37-78-00

GERMANY

Phone: +49-6102-3689-0

HONG KONG

Phone: +852-3104-1010

HUNGARY

Phone: +36 1 802 7360

INDIA

Phone: +91-44-4963-0900

INDONESIA

Phone: +62-21-2966-0120

ITALY

Phone: +39-02-6688220

KOREA

Phone: +82-31-789-4300

MALAYSIA

Phone: +60-3-7883-2211

MEXICO

Phone: +52-55-8850-0100

NETHERLANDS

Phone: +31 40 20 66 100

PHILIPPINES

Phone: +63-2-8981-5000

POLAND

Phone: +48 71 36861 60

ROMANIA

Phone: +40 269 232 808

SINGAPORE

Phone: +65-6392-1011

SLOVAKIA

Phone: +421 2 5939 6461

SLOVENIA

Phone: +386 1 4701 666

SWITZERLAND

Phone: +41 43 455 77 30

TAIWAN

Phone: +886-2-2721-8080

THAILAND

Phone: +66-2-369-2777

UK & IRELAND

Phone: +44 1908-696-900

USA

Phone: +1-201-930-0100

VIETNAM

Phone: +84-24-3772-5555

