MIL Command Listing and Description

This section provides an overview of each MIL module and a brief description of each MIL command. For a complete description of the syntax and use of each command, refer to the MIL Command Reference manual.

1D and 2D Code Reader module

Used to read (and write) various 1D and 2D code symbologies.

Commands	Command parameters	Description
McodeAlloc()	SystemId, CodeType, ControlFlag, CodeIdPtr	Allocate a code object.
McodeControl()	CodeId, ControlType, ControlValue	Control a code object.
McodeFree()	CodeId	Free a code object.
McodeGetResult()	CodeId, ResultType, ResultPtr	Get a result from a read or write operation.
McodeInquire()	CodeId, InquireType, UserVarPtr	Inquire about a code object parameter setting.
McodeRead()	CodeId, ImageBufId, ControlFlag	Read a specific type of code in an image.
McodeRestore()	FileName, SystemId, ControlFlag, CodeIdPtr	Restore a code object previously saved to a file.
McodeSave()	FileName, CodeId, ControlFlag	Save the specified code object in a file.
McodeStream()	MemPtrOrFileName, SystemId Operation, StreamType, Version ControlFlag, CodeIdPtr, SizeByteVarPtr	Load, restore, or save a code object from/to a file or memory.
McodeVerify()	CodeId, ImageBufId, String, ControlFlag	Compute the different quality-grades of the code in the specified source image.
McodeWrite()	CodeId, ImageBufId, String, ControlFlag	Encode an ASCII string.

1D and 2D code symbologies

For the McodeAlloc() command, the code type(s) that can be read or written include(s):

Code Type	Encoding Type	Error Correction
BC412	Standard encoding type	No error correction
Codabar	Standard encoding type	No error correction
Code39	ASCII encoding, Standard encoding type	No error correction; check-digit error correction
Code93	ASCII encoding	Check-digit error correction
Code128 (UCC/EAN128)	ASCII encoding	Check-digit error correction

Continued...

1D and 2D code symbologies (cont.)

For the McodeAlloc() command, the code type(s) that can be read or written include(s):

Code Type	Encoding Type	Error Correction
DataMatrix	Numeric encoding, Alpha encoding, AlphaNumericPunc encoding, AlphaNumeric encoding, ASCII encoding, IS08 encoding	10, 40, 50, 60, 70, 80,90, 100, 110, 120, 130, 140 or 200 error correction
EAN8	Numeric encoding	Check-digit error correction
EAN13	Numeric encoding	Check-digit error correction
Interleaved 2/5	Numeric encoding	No error correction; check-digit error correcion
Maxicode	Encoding mode 2, 3, 4, 5, 6	Reed Solomon error correction
MicroPDF417	Standard encoding type	Reed Solomon error correction
PDF417	Standard encoding type	Reed Solomon 1 - 8 error correction
Pharma	Numeric encoding	No error correction
Planet	Numeric encoding	Check-digit error correction
Postnet	Numeric encoding	Check-digit error correction
QR	QR code Model 1, 2 encoding	Lowest-level QR, Low-level QR, High-level QR, Highest-level QR
RSS	RSS 14, RSS 14 Stacked, RSS 14 Stacked Omni, RSS 14 Truncated, RSS Expanded RSS Expanded Stacked, RSS Limited encoding.	Check-digit error correction
UPC-A	Numeric encoding	Check-digit error correction
UPC-E	Numeric encoding	Check-digit error correction

Composite code symbologies

This code type is a composite of a 1D (RSS, UPC-A, UPC-E, EAN-8, EAN-13, or UCC/EAN128) and a 2D code type (PDF417 or MicroPDF417).

Application and System modules

Used to initialize and control the MIL application environment and system (frame grabber boards, vision processor boards, or host system) respectively. The Application module includes control of integrated debugging features, system resource compensation, command threads and related events, as well as a timer function.

Commands	Command parameters	Description
MappAlloc()	InitFlag, ApplicationIdPtr	Allocate a MIL application.
MappAllocDefault()	InitFlag, ApplicationIdPtr, SystemIdPtr, DisplayIdPtr, DigIdPtr, ImageBufIdPtr	Allocate MIL application defaults.
MappControl()	ControlType, ControlFlag	Control an application environment setting.
MappFree()	ApplicationId	Free a MIL application.
MappFreeDefault()	ApplicationId, SystemId, DisplayId, DigId, ImageBufId	Free MIL application defaults.
MappGetError()	ErrorType, ErrorPtr	Get error codes and related information.
MappGetHookInfo()	EventId, InfoType, UserVarPtr	Get information about a hooked event.
MappHookFunction()	HookType, HookHandlerPtr, ExpansionFlag	Hook a function to an event.
MappInquire()	InquireType, UserVarPtr	Inquire about the application parameter setting.
MappTimer()	ControlValue, TimePtr	Control the MIL timer.
MsysAlloc()	SystemTypePtr, SystemNum, InitFlag, SystemIdPtr	Allocate a hardware system.
MsysControl()	SystemId, ControlType, ControlFlag	Control system behavior.
MsysFree()	SystemId	Free a system.
MsysHookFuntion()	SystemId, HookType, HookHandlerPtr, UserDataPtr	Hook a function to a system event.
MsysInquire()	SystemId, ParamToInquire, UserVarPtr	Inquire about a system parameter setting.

Blob analysis module

Used to identify and measure connected components (blobs) in an image.

Commands	Command parameters	Description
MblobAllocFeatureList()	SystemId, FeatureListIdPtr	Allocate a blob analysis feature list.
MblobAllocResult()	SystemId, BlobResIdPtr	Allocate a blob analysis result buffer.
MblobCalculate()	BlobIdentImageId, GrayImageId, FeatureListId, BlobResId	Perform blob analysis calculations.
MblobControl()	BlobResId, Procmode, Value	Control a blob analysis processing mode setting.
MblobDraw()	GraphContId, ResultId, DestImageId, Operation, Label, ControlFlag	Draw features of specified blob results in an image buffer.
MblobFill()	BlobResId, DestImageBufId, Criteria, Value	Draw blobs that meet a specified fill criterion.
MblobFree()	BlobId	Free the blob analysis result buffer or the feature list.
MblobGetLabel()	BlobResId, XPos, YPos, LabelVarPtr	Get the label value of a blob at a specific position.
MblobGetNumber()	BlobResId, CountVarPtr	Get the number of currently included blobs.
MblobGetResult()	BlobResId, Feature, TargetArrayPtr	Read feature values of the included blobs.
MblobGetResultSingle()	BlobResId, LabelVal, Feature, TargetArrayPtr	Read the feature value of a single blob.
MblobGetRuns()	BlobResId, LabelVal, ArrayType, RunXPtr, RunYPtr, RunLengthPtr	Get the blob run-length encoding information.
MblobInquire()	BlobResId, InquireType, UserVarPtr	Inquire about a blob analysis processing mode.
MblobLabel()	BlobResId, DestImageBufId, Mode	Draw a labeled image.
MblobReconstruct()	SrcImageBufId, SeedImageBufId, DestImageBufId, Operation, ProcMode	Reconstruct blobs (or blob holes) in an image buffer.
MblobSelect()	BlobResId, Operation, Feature, Condition, CondLow, CondHigh	Select blobs for calculations and result retrieval.
MblobSelectFeature()	FeatureListId, Feature	Select feature(s) to be calculated. See complete feature list on the following page.
MblobSelectFeret()	FeatureListId, Angle	Add Feret angle to the feature list.
MblobSelectMoment()	FeatureListId, MomType, XMomOrder, YMomOrder	Add specified moment calculations to the feature list.

Blob features

For the MblobSelectFeature() command, the feature(s) that can be calculated include(s):

M_AREA, the number of foreground pixels in a blob.
M_BOX_X_MIN, M_BOX_Y_MIN, M_BOX_X_MAX, M_BOX_Y_MAX, the coordinates of the extreme left, top, right and bottom pixels, respectively, of a blob.
M_BREADTH, a measure of the true breadth of an object.
M_CHAIN_INDEX, this is the index which differentiates chains in a blob.
M_CHAIN_Y, M_CHAIN_X, these are the x and y coordinates of each chained pixel.
M_COMPACTNESS, a minimum for a circle (1.0) and is derived from the perimeter and area.
M_CONVEX_PERIMETER, an approximation of the perimeter of the convex hull of a blob.
M_ELONGATION, equal to M_LENGTH over M_BREADTH.
M_EULER_NUMBER, the number of blobs minus the number of holes.
M_FERET_X, M_FERET_Y, the dimensions of the minimum bounding box of a blob in the horizontal and vertical directions (respectively).
M_FERET_MIN_DIAMETER, the smallest Feret diameter found after checking a certain number of angles.
M_FERET_MIN_ANGLE, the angle at which the minimum Feret diameter is found.
M_FERET_MAX_DIAMETER, the largest Feret diameter found after checking a certain number of angles.
M_FERET_MAX_ANGLE, the angle at which the maximum Feret diameter is found.
M_FERET_MEAN_DIAMETER, the average Feret diameter at all the angles checked.
M_FERET_ELONGATION, a measure of the shape of a blob.
M_FIRST_POINT_X, M_FIRST_POINT_Y, a unique point for each object, which is always on the perimeter of the object.
M_INTERCEPT _0, _45, _90, _135, the number of times that a transition from background to foreground occurs at the given angle for the entire blob.
M_LABEL_VALUE, the label value for each blob in an image.
M_LENGTH, a measure of the true length of an object.
M_NUMBER_OF_CHAINED_PIXELS, this is the number of chained pixels for all blobs or a specified blob.
M_NUMBER_OF_HOLES, the number of holes in a blob.
M_NUMBER_OF_RUNS, the total number of horizontal strings of consecutive foreground pixels in a blob.
M_PERIMETER, the total length of edges in a blob (including the edges of any holes).
M_ROUGHNESS, a measure of how rough a blob is.
M_X_MIN_AT_Y_MIN, M_X_MAX_AT_Y_MAX, M_Y_MIN_AT_X_MAX, M_Y_MAX_AT_X_MIN, these values, together with the four box coordinates, give four contact points on the convex perimeter of the object.

For a grayscale image:
M_MEAN_PIXEL, the mean pixel value in a blob.
M_MIN_PIXEL, the minimum pixel value found in a blob.
M_MAX_PIXEL, the maximum pixel value found in a blob.
M_SIGMA_PIXEL, the standard deviation of pixel values in a blob.
M_SUM_PIXEL, the sum of all pixel values in a blob.
M_SUM_PIXEL_SQUARED, the sum of the squares of each pixel value in a blob.

The following features have two different definitions: a binary one, where all pixels are considered equal; and a grayscale one, where pixels are weighted by their value in the gray scale image.
M_CENTER_OF_GRAVITY_X, the x position of the center of gravity of a blob.
M_CENTER_OF_GRAVITY_Y, the y position of the center of gravity of a blob.
M_MOMENT_Xn_Ym and M_MOMENT_CENTRAL_Xn_Ym for central moments; coordinates are relative to each blob's center of gravity; ordinary moments use coordinates relative to the image origin.
M_AXIS_PRINCIPAL_ANGLE, the angle at which a blob has the least moment of inertia.
M_AXIS_SECONDARY_ANGLE, the angle perpendicular to M_AXIS_PRINCIPAL_ANGLE.

The following predefined values let the user select groups of features in a single call:
M_BOX, adds all 4 box features plus x and y Ferets.
M_CONTACT_POINTS, adds first point and other contact features.
M_CENTER_OF_GRAVITY, adds both x and y coordinates of the center of gravity.
M_ALL_FEATURES, adds all features (except general Feret and general moment).
M_NO_FEATURES, removes all features (except label value).
M_CHAINS, adds all 4 chain features.

You can add the following sorting options to a feature to specify it as a sorting key during result retrieval:
M_SORTn_DOWN, specifies the feature as the nth sorting key (in a descending order) where n is an integer between 1 and 3.
M_SORTn_UP, specifies the feature as the nth sorting key (in an ascending order) where n is an integer between 1 and 3.
M_NO_SORT, removes the specified sorting key.

Buffer and Data generation modules

Used to allocate and control a data buffer, and to generate data for the LUT and the warp function. The Buffer module includes control of a child buffer (ROI), buffer compression and decompression, custom kernel or structuring element, and buffer archiving and retrieving.

Commands	Command parameters	Description
MbufAlloc1d()	SystemId, SizeX, Type, Attribute, BufIdPtr	Allocate a 1D data buffer.
MbufAlloc2d()	SystemId, SizeX, SizeY, Type, Attribute, BufIdPtr	Allocate a 2D data buffer.
MbufAllocColor()	SystemId, SizeBand, SizeX, SizeY, Type, Attribute, BufIdPtr	Allocate a color data buffer.
MbufBayer()	SrcImageBufId, DestImageBufId, WhiteBalanceCoefficientsID, ControlFlag	Decode the color information of a single-band, Bayer color-encoded image.
MbufChild1d()	ParentBufId, OffX, SizeX, BufIdPtr	Allocate a 1D child data buffer.
MbufChild2d()	ParentBufId, OffX, OffY, SizeX, SizeY, BufIdPtr	Allocate a child buffer from a specific region of the parent buffer.
MbufChildColor()	ParentBufId, Band, BufIdPtr	Allocate a color-band child data buffer within a color parent buffer.
MbufChildColor2d()	ParentBufId, Band, OffX, OffY, SizeX, SizeY, BufIdPtr	Allocate a child data buffer within a color parent buffer.
MbufChildMove()	BufferID, OffsetX, OffsetY,SizeX, SizeY ControlFlag	Move and resize a child buffer within the parent buffer
MbufClear()	DestImageBufId, Color	Clears a buffer to a specified color.
MbufControl()	BufId, ControlType, ControlValue	Control specified buffer features.
MbufControlNeighborhood()	BufId, OperationFlag, OperationValue	Change the value of an operation flag associated with a custom kernel or structuring element.
MbufControlRegion()	BufId, OffsetX, OffsetY, SizeX, SizeY, Band, ControlType, ControlValue,	Control a specified region of a buffer.
MbufCopy()	SrcBufId, DestBufId	Copy data from one buffer to another (optionally with compression or format conversion).
MbufCopyClip()	SrcBufId, DestBufId, DestOffX, DestOffY	Copy buffer, clipping data outside destination buffer.
MbufCopyColor()	SrcBufId, DestBufId, Band	Copy one or all bands of an image buffer.
MbufCopyColor2d()	SrcBufId, DestBufId, ScrBand, ScrOffX, ScrOffY, DstBand, DstOffX, DstOffY, SizeX, SizeY	Copy a 2D region of one or all bands of an image buffer to another buffer.
MbufCopyCond()	SrcBufId, DestBufId, CondBufId, Condition, CondValue	Copy conditionally the source buffer to the destination buffer.
MbufCopyMask()	SrcBufId, DestBufId, MaskValue	Copy buffer with mask.
MbufCreateColor()	SystemId, SizeBand, SizeX, SizeY, Type, ControlFlag, Pitch, ArrayOfDataPtr BufIdPtr	Create a color data buffer.
MbufCreate2d()	SystemId, SizeX, SizeY, Type, Attribute, ControlFlag, Pitch, DataPtr, BufIdPtr	Create a two-dimensional data buffer.
MbufDiskInquire()	FileName, ParamToInquire, UserVarPtr	Inquire about the buffer data in a file.
MbufExport()	FileName, FileFormat, SrcBufId	Export a data buffer to a file.
MbufExportSequence()	FileName, FileFormatId, BufArrayPtr, NumberOfImages,FrameRate, ControlFlag	Export a sequence of image buffers to an AVI file.
MbufFree()	BufId	Free a data buffer.
MbufGet1d()	SrcBufId, OffX, SizeX, UserArrayPtr	Get data from a 1D area of a buffer and place it in a user-supplied array.

Buffer and Data generation modules (continued)

Commands	Command parameters	Description
MbufGet2d()	SrcBufId, OffX, OffY, SizeX, SizeY, UserArrayPtr	Get data from a 2D area of a buffer and place it in a user-supplied array.
MbufGet()	SrcBufId, UserArrayPtr	Get data from a buffer and place it in a user-supplied array.
MbufGetArc()	ImageBufId, XCenter, YCenter, XRad YRad, StartAngle, EndAngle, NbPixelsPtr, UserArrayPtr	Read the pixels along a specified arc and store their values in a user-defined array.
MbufGetColor()	SrcBufId, DataFormat, Band, UserArrayPtr	Get data from one or all bands of a buffer and place it in a user-supplied array.
MbufGetColor2d()	SrcBufId, DataFormat, Band, OffX, OffY, SizeX, SizeY, UserArrayPtr	Get data from a region of one or all bands of a buffer and place it in a user-supplied array.
MbufGetHookInfo()	EventId, InfoType, UserVarPtr	Get information about a hook event.
MbufHookFunction()	BufferId, HookType, HookHandlerPtr, UserDataPtr	Hook a function to a buffer event.
MbufGetLine()	ImageBufId, StartX, StartY, EndX, EndY, Mode, NumPixelsPtr, UserArrayPtr	Read the pixels of a theoretical line between specified coordinates, count them, and store them in a user-defined array.
MbufImport()	FileName, FileFormat, Operation, SystemId, BufIdPtr	Import data from a file into a data buffer.
MbufImportSequence()	FileName, FileFormatId, Operation, SystemId, BufArrayPtr, StartImage, NumberOfImages, ControlFlag	Import a sequence of images from an AVI file into separate image buffers.
MbufInquire()	BufId, ParamToInquire, UserVarPtr	Inquire about a data buffer parameter setting.
MbufLoad()	FileName, BufId	Load data from a file into a data buffer.
MbufPut()	DestBufId, UserArrayPtr	Put data from a user-supplied array into a data buffer.
MbufPutColor()	DestBufId, DataFormat, Band, UserArrayPtr	Put data from a user-supplied array into one or all bands of a data buffer.
MbufPutColor2d()	DestBufId, DataFormat, Band, OffX, OffY, SizeX, SizeY, UserArrayPtr	Put data from a user-supplied array into a region of one or all bands of a data buffer.
MbufPutLine()	ImageBufId, StartX, StartY, EndX, EndY, Mode, NumbPixelsPtr, UserArrayPtr	Write a specified series of pixels within specified coordinates, along a theoretical line.
MbufPut1d()	DestBufId, OffX, SizeX, UserArrayPtr	Put data from a user-supplied array into a 1D area of a buffer.
MbufPut2d()	DestBufId, OffX, OffY, SizeX, SizeY, UserArrayPtr	Put data from a user-supplied array into a 2D area of a buffer.
MbufRestore()	FileName, SystemId, BufIdPtr	Restore data from a file into an automatically allocated data buffer.
MbufSave()	FileName, BufId	Save a data buffer in a file, using the MIL output file format.
MbufTransfer()	SrcBufId, DestBufId, SrcOffX, SrcOffY SrcBand, DestOffX, DestOffY, DestSizeX DestSizeY DestBand, TransferFunction TransferType, OperationFlag ExtraParameter	Copy a 2D region of one or all bands from the source buffer into a 2D region of one or all bands in the destination buffer, using a specified transfer function and transfer type file format.
MgenLutFunction()	LutBufId, Func, a, b, c, StartIndex, StartXValue, EndIndex	Generate data into a LUT buffer using a specified standard mathematical function.
MgenLutRamp()	LutId, StartIndex, StartValue, EndIndex, EndValue	Generate ramp data into a LUT buffer.
MgenWrapParameters()	InWarpParameter, OutXLutOrCoef, OutYLut, OperationMode, Transform, Val1, Val2	Generate coefficients or LUTs for use with MimWarp().

Calibration module

Used to convert coordinates or measurements from pixel to real-world units, as well as to correct distortions in an image.

Commands	Command parameters	Description
McalAlloc()	Mode, ModeFlag, CalibrationIdPtr	Allocate a calibration object.
McalAssociate() to/from	CalibrationId, ImageOrDigitizerId,	Associate/disassociate a calibration object ControlFlag an image or digitizer.
McalControl()	CalibrationId, ControlType, ControlValue	Control a calibration object parameter setting.
McalFree()	CalibrationId	Free a calibration object.
McalGrid()	CalibrationId, SrcImageBufId, GridOffsetX, GridOffsetY, GridOffsetZ, RowNumber, ColumnNumber, RowSpacing, ColumnSpacing, Mode, ModeFlag	Calibrate your imaging setup using a grid.
McalInquire()	CalibrationOrMilId, InquireType, UserVarPtr	Inquire about a calibration object setting or about the calibration object associated to an image or digitizer.
McalList()	CalibrationId, XPixArray, YPixArray, XWorldArray, YWorldArray, ZWorld, NumPoint, Mode, ModeFlag	Calibrate your imaging setup using a list of coordinates.
McalRelativeOrigin()	CalibrationId, XOffset, YOffset, ZOffset, AngularOffset, ControlFlag	Change the origin and/or orientation of a relative coordinate system.
McalRestore()	FileName, ControlFlag, CalibrationIdPtr	Restore a calibration object from a file.
McalSave()	FileName, CalibrationId, ControlFlag	Save a calibration object to a file.
McalStream()	MemPtrOrFileName, SystemId, Operation, StreamType, Version, ControlFlag, CalibrationIdPtr SizeByteVarPtr	Load, restore, or save a calibration object from/to a file or a memory.
McalTransformCoordinate()	CalibrationOrMilId, TransformType, X, Y, ResXPtr, ResYPtr	Convert coordinates between world and pixel values.
McalTransformCoordinateList()	CalibrationOrMilId, TransfromType, NumPoints, SrcXPtr, SrcYPtr, ResXPtr, ResYPtr	Convert a list of coordinates between their world and pixel values.
McalTransformImage()	SrcImageBufId, DestImageBufId, CalibrationId, InterpolationMode, OperationType, ControlFlag	Physically transform an image to remove any distortions.
McalTransformResult()	CalibrationOrMilId, TransformType, ResultType, Result, ResResult	Convert a result between world and pixel value.

Digitizer module

Used to initialize and control a digitizer (image capture device). This module includes control of capture mode (trigger, frame/field, blocking/non-blocking), image scaling and cropping, input channel, input LUT, analog settings (references, hue, saturation, and brightness) as well as events for callback functions.

Commands	Command parameters	Description
MdigAlloc()	SystemId, DigNum, DataFormat, InitFlag, DigIdPtr	Allocate a digitizer.
MdigChannel()	DigId, Channel	Select the active input channel of a digitizer.
MdigControl()	DigId, ControlType, Value	Control the specified digitizer.
MdigFocus()	DigId, DestImageBufId, FocusImageRegionBufId, FocusHookPtr, UserDataPtr, MinPosition, StartPosition, MaxPosition, MaxPositionVariation, ProcMode, ResultPtr	Adjust a camera's lens motor to a position which provides optimum focus.
MdigFree()	DigId	Free a digitizer.
MdigGrab()	ScrDigId, DestImageBufId	Grab data from an input device into a buffer.
MdigGrabContinuous()	DigId, DestImageBufId	Grab data continuously from an input device.
MdigGrabWait()	DigId, Flag	Wait for the end of the grab in progress.
MdigHalt()	DigId	Halt a continuous grab from an input device.
MdigHookFunction()	DigId, HookType, HookHandlerPtr, UserDataPtr	Hook a function to a digitizer event.
MdigInquire()	DigId, InquireType, UserVarPtr	Inquire about a digitizer parameter setting.
MdigLut()	DigId, LutBufId	Copy a LUT buffer to a digitizer LUT.
MdigProcess()	DigId, DestImageArrayPtr, ImageCount Operation, OperationFlag, HookHandlerPtr, UserDataPtr	Grabs a sequence of images and process them with a user-defined function as they are grabbed.
MdigReference()	DigId, ReferenceType, ReferenceLevel	Select digitization reference level.

Display module

Used to initialize and control an image display. This module includes control of image display windows, graphics overlay, output LUT, image pan, scroll, and zoom.

Commands	Command parameters	Description
MdispAlloc()	SystemId, DispNum, DispFormat, InitFlag, DisplayIdPtr	Allocate a display.
MdispControl()	DisplayId, ControlType, ControlValue	Control the MIL display.
MdispFree()	DisplayId	Free a display.
MdispHookFunction()	DisplayId, HookType, HookHandlerPtr, UserDataPtr	Hook a function to a display event.
MdispInquire()	DisplayId, InquireType, UserVarPtr	Inquire about a display parameter setting.
MdispLut()	DisplayId, LutBufId	Copy a LUT buffer to a display output LUT.
MdispPan()	DisplayId, XOffset, YOffset	Pan and scroll a display.
MdispSelect()	DisplayId, ImageBufId	Select an image buffer to display.
MdispSelectWindow()	DisplayId, ImageBufId, ClientWindowHandle	Select an image buffer to display in a user-defined window.
MdispZoom()	DisplayId, XFactor, YFactor	Zoom a display.

Edge Finder module

Used to extract and analyze object contours or thin curvilinear features.

Commands	Command parameters	Description
MedgeAlloc()	SystemId, EdgeFinderType, ControlFlag, ContextIdPtr	Allocate an Edge Finder context.
MedgeAllocResult()	SystemId, ControlFlag, EdgeResultIdPtr,	Allocate an Edge Finder result buffer.
MedgeCalculate()	ContextId, SourceImageId, SourceDeriv1Id, SourceDeriv2Id, SourceDeriv3Id, EdgeResultId, ControlFlag	Perform edge extraction and feature calculations.
MedgeControl()	ContextOrResultId, ControlType, ControlValue	Control an Edge Finder context or an Edge Finder result buffer.
MedgeDraw()	GraphContId, EdgeResultId, DestImageId, Operation, IndexOrLabel, ControlFlag	Draw specific edge features in the destination image buffer.
MedgeFree()	ObjectId	Free an Edge Finder context or an Edge Finder result buffer.
MedgeGetNeighbors()	EdgeResultId, SizeOfArray, SrcArrayXPtr SrcArrayYPtr, SrcArrayAnglePtr, DstArrayXPtr, DstArrayYPtr, DstArrayIndexPtr, DstArrayLabelPtr, ControlFlag	Get edgels from an Edge Finder result buffer that are the closest neighbors to a list of user-specified point coordinates.
MedgeGetResult()	EdgeResultId, EdgeIndexOrLabelValue, ResultType, FirstResultArrayPtr, SecondResultArrayPtr	Get results of the included edges from an Edge Finder result buffer.
MedgeInquire()	ContextOrResultId, InquireType, UserVarPtr	Inquire about an Edge Finder context or an Edge Finder result buffer.
MedgeMask()	ContextId, MaskImageId, ControlFlag	Mask regions of the image.
MedgeRestore()	Filename, SystemId, ControlFlag, ContextIdPtr	Restore an Edge Finder context from disk.
MedgeSave()	FileName, ContextOrResultId, ControlFlag	Save an Edge Finder context to a file, or save edge chains and/or edge approximations from an Edge Finder result buffer to a CAD (Computer-Aided Design) file.
MedgeSelect()	EdgeResultId, Operation, Feature, Condition, Param1, Param2	Select edges for calculations and result retrieval.
MedgeStream()	MemPtrOrFileName, SystemId, Operation, StreamType, Version, ControlFlag, ContextOrResultIdPtr, SizeByteVarPtr	Load, restore, or save an Edge Finder context from/to a file or memory, or save calculated edges from an Edge Finder result buffer to a file or memory in DXF format.

Edge features

For the MedgeGetResults() command, the feature(s) that can be calculated include(s):

M_AVERAGE_STRENGTH, returns the average strength value of each edge.

M_BOX_X_MAX, returns the X-coordinate of each edge's right-most edgel.

M_BOX_X_MIN, returns the X-coordinate of each edge's left-most edgel.

M_BOX_Y_MAX, returns the Y-coordinate of each edge's bottom-most edgel.

M_BOX_Y_MIN, returns the Y-coordinate of each edge's top-most edgel.

M_BULGES, returns the bulge values between vertices.

M_CENTER_OF_GRAVITY, returns the coordinates of each edge's center of gravity.

M_CENTER_OF_GRAVITY_X, returns the X-coordinate of each edge's center of gravity.

M_CENTER_OF_GRAVITY_Y, returns the Y-coordinate of each edge's center of gravity.

M_CIRCLE_FIT_CENTER_X, returns the X-coordinate of the center of the circle that is the best fit for each edge.

M_CIRCLE_FIT_CENTER_Y, returns the Y-coordinate of the center of the circle that is the best fit for each edge.

Continued...

Edge features (continued)

M_CIRCLE_FIT_COVERAGE, returns the coverage of the circle that is the best fit for each edge.

M_CIRCLE_FIT_ERROR, returns the fit error of the circle that is the best fit for each edge.

M_CIRCLE_FIT_RADIUS, returns the radius of the circle that is the best fit for each edge.

M_CHAIN, returns the coordinates of the edge(s)'s edgels.

M_CHAIN_ANGLE, returns the direction of the edge(s)'s edgels.

M_CHAIN_CODE, returns the edge(s)'s chain code.

M_CHAIN_INDEX, returns the index of the edge(s)'s edgels.

M_CHAIN_MAGNITUDE + M_CHAIN_ANGLE, returns the magnitude values and the angle values of the edge(s)'s edgels.

M_CHAIN_MAGNITUDE, returns the magnitude values of the edge(s)'s edgels.

M_CHAIN_X, Y, returns the X or Y-coordinates of the edge(s)'s edgels.

M_CLOSURE, Returns the closure of each edge.

M_CONVEX_PERIMETER, returns the convex elongation of each edge.

M_ELLIPSE_FIT_ANGLE , returns the angle of the ellipse that is the best fit for each edge.

M_ELLIPSE_FIT_CENTER_X, Y, returns the X or Y-coordinate of the center of the ellipse that is the best fit for each edge.

M_ELLIPSE_FIT_COVERAGE, returns the coverage of the ellipse that is the best fit for each edge.

M_ELLIPSE_FIT_ERROR, returns the fit error of the ellipse that is the best fit for each edge.

M_ELLIPSE_MAJOR_AXIS, returns the major axis of the ellipse that is the best fit for each edge.

M_ELLIPSE_MINOR_AXIS, returns the minor axis of the ellipse that is the best fit for each edge.

M_FAST_LENGTH, returns the fast length of each edge.

M_FERET_BOX, returns the X- and Y-Feret values of each edge.

M_FERET_ELONGATION, returns the Feret elongation of each edge.

M_FERET_MAX_ANGLE, returns the maximum Feret angle of each chain, in degrees.

M_FERET_MAX_DIAMETER, returns the maximum Feret diameter of each edge.

M_FERET_MEAN_DIAMETER, returns the average Feret diameter at all the angles checked.

M_FERET_MIN_ANGLE, returns the minimum Feret angle of each chain.

M_FERET_MIN_DIAMETER, returns the minimum Feret diameter of each edge.

M_FERET_X, Y, returns the X or Y-Feret value of each edge.

M_FIRST_POINT, returns the coordinates of each edge's first point.

M_FIRST_POINT_X, Y, returns the X or Y-coordinate of each edge's first point.

M_GENERAL_FERET, returns the general Feret of each edge.

M_LABEL_VALUE, returns the label value of each edge in an image.

M_LENGTH, returns the length of each edge.

M_LINE_FIT_A , _B, _C, returns the A, B or C variable of the line that is the best fit for each edge.

M_LINE_FIT_ERROR, returns the fit error of the line that is the best fit for each edge.

M_MOMENT_ELONGATION, returns the moment elongation of each edge.

M_MOMENT_ELONGATION_ANGLE, returns the angle of the principle axis along each edge's moment elongation.

M_NUMBER_OF_CHAINED_EDGELS, returns the total number of edgels in the edge(s).

M_NUMBER_OF_CHAINS, returns the number of included edges.

M_NUMBER_OF_VERTICES, returns the total number of chain approximation vertices in the edge(s).

M_POSITION, returns the X- and Y-position of each edge.

M_POSITION_X, Y, returns the X or Y-position of each edge.

M_SIZE, returns the number of edgels in each edge.

M_STRENGTH, returns the strength value of each edge.

M_TORTUOSITY, returns the tortuosity measure of each edge.

M_VERTICES, returns the coordinates of the chain approximation's vertices.

M_VERTICES_X, Y, returns the X or Y-coordinates of the chain approximation's vertices.

M_X_MAX_AT_Y_MAX, returns the maximum X-coordinate at the maximum Y-coordinate of each edge.

M_X_MIN_AT_Y_MIN, returns the minimum X-coordinate at the minimum Y-coordinate of each edge.

M_Y_MAX_AT_X_MIN, returns the maximum Y-coordinate at the minimum X-coordinate of each edge.

M_Y_MIN_AT_X_MAX, returns the minimum Y-coordinate at the maximum X-coordinate of each edge.

Function Developer's Toolkit

The MIL Function Developer's Toolkit allows programmers to define functions to extend MIL's functionality. Using this toolkit, you can implement functions and integrate them directly into the MIL library, where they behave like standard MIL functions (e.g., respecting error handling and tracing).

Commands	Command parameters	Description
MfuncAlloc()	FunctionName, ParameterNumber, SlaveFunctionPtr, Reserved1, Reserved2, SlaveFunctionOpcode, InitFlag, FuncIdPtr	Allocate a MIL function context for your user-defined function.
MfuncAllocId()	FunctionId, ObjectType, ObjectPtr	Associate a MIL identifier with a user-defined object.
MfuncCall()	FunctionId	Execute the slave function.
MfuncErrorReport()	FunctionId, ErrorCode, ErrorMessage, ErrorSubMessage1, ErrorSubMessage2, ErrorSubMessage3	Report an error message.
MfuncFree()	FunctionId	Free a MIL function context.
MfuncFreeId()	FunctionId, ObjectId	Free the MIL identifier associated with a user-defined MIL object.
MfuncInquire()	ObjectId, InquireType, UserVarPtr	Retrieve information on a user-defined MIL object.
MfuncParamCheck()	FunctionId	Verify whether parameter checking is required.
MfuncParamDouble()	FunctionId, ParamIndex, ParamValue	Register a parameter of type double.
MfuncParamId()	FunctionId, ParamIndex, ParamValue ParamIs, RequiredAttribute	Register a MIL_ID parameter.
MfuncParamIdPointer()	FunctionId, ParamIndex, ParamValue, ParamIs, ParamAttribute	Register a MIL_ID pointer parameter
MfuncParamLong()	FunctionId, ParamIndex, ParamValue	Register a parameter of type long.
MfuncParamPointer()	FunctionId, ParamIndex, ParamValue Size, Attribute	Register a pointer parameter.
MfuncParamString()	FunctionId, ParamIndex, ParamValue Size, Attribute	Register a null-terminated string parameter.
MfuncParamValue()	FunctionId, ParamIndex, ParamValuePtr	Read the value of the specified MIL function parameter.

Geometric Model Finder module

Use geometric features (i.e., contours) to find models in an image. This module includes functions to define models, control search strategy, and save and restore models.

Commands	Command parameters	Description
MmodAlloc()	SystemId, ModelFinderType, ControlFlag, ContextIdPtr	Allocate a model finder context.
MmodAllocResult()	SystemId, ControlFlag, ModResultIdPtr	Allocate a model finder result buffer.
MmodControl()	ContextId, Index, ControlType, ControlValue	Control a model finder context setting.
MmodDefine()	ContextId, ModelType, Param1, Param2, Param3, Param4, Param5	Add a model to, or delete model from, a model finder context.
MmodDefineFromFile()	ContextId, FileType, Filename, ControlFlag	Defines a model from a file and adds it to a Model Finder context.
MmodDraw()	GraphContId, ContextOrResultId, DestImageId, Operation, Index, ControlFlag	Draw features of specific models or result occurrences in an image buffer.
MmodFind()	ContextId, TargetImageId, ModResultId	Search for the model(s) of the specified Model Finder context in a target image buffer or in an Edge Finder result buffer.
MmodFree()	ObjectId	Free a measurement context, marker, or result buffer.
MmodGetResult()	ResultId, ResultIndex, ResultType, ResultArrayPtr	Get the model finder result values.
MmodInquire()	ImageBufId, ModelId, FindResultId, ResultRange	Inquire information from a specified model finder context.
MmodMask()	ContextId, Index, MaskBufferId, MaskType, ControlFlag	Mask regions of a model result buffer.
MmodPreprocess()	ContextId, ControlFlag	Preprocess a model finder context.
MmodRestore()	FileName, SystemId, ControlFlag, ContextIdPtr	Restore a model finder context from disk.
MmodSave()	FileName, ContextId, ControlFlag	Save a model finder context to a file.
MmodStream()	MemPtrOrFileName, SystemId, Operation, StreamType, Version ControlFlag, ContextIdPtr, SizeByteVarPtr	Load, restore, or save a Model Finder context from/to a file or a memory.

Graphics module

Used to create drawings and text annotations in an image. This module provides a set of graphics primitives (arc, circle, line, and rectangle), control of color (foreground, background, fill), and text (font, color, size).

Commands	Command parameters	Description
MgraAlloc()	SystemId, GraphContIdPtr	Allocate a graphics context.
MgraArc()	GraphContId, DestImageBufId, XCenter, YCenter, XRad, YRad, StartAngle, EndAngle	Draw an arc.
MgraArcFill()	GraphContId, DestImageBufId, XCenter, YCenter, XRad, YRad, StartAngle, EndAngle	Draw a filled elliptic arc.
MgraBackColor()	GraphContId, BackgroundColor	Sets the background color of a graphics context.
MgraClear()	GraphContId, DestImageBufId	Clear an image buffer to a specified foreground color.
MgraColor()	GraphContId, ForegroundColor	Sets the foreground color of a graphics context.
MgraControl()	GraphContId, ControlType, Control	Control the specified graphics context.
MgraDot()	GraphContId, DestImageBufId, XPos, YPos	Draw a dot.
MgraDots()	GraphContId, DestImageBufId, NumberOfDots, XPosArray, YPosArray, ControlFlag	Draw one or more dots
MgraFill()	GraphContId, DestImageBufId, XStart, YStart	Perform a boundary-type seed fill.
MgraFont()	GraphContId, FontName	Associate a text font with a graphics context.
MgraFontScale()	GraphContId, XFontScale, YFontScale	Set the font scale of a graphics context.
MgraFree()	GraphContId	Free a graphics context.
MgraInquire()	GraphContId, InquireType, UserVarPtr	Inquire about the graphics parameters.
MgraLine()	GraphContId, DestImageBufId, XStart, YStart, XEnd, YEnd	Draw a line.
MgraLines()	GraphContId, DestImageBufId, NumberOfLines, XStartArray, YStartArray, XEndArray, YEndArray ControlFlag	Draw one or more lines.
MgraRect()	GraphContId, DestImageBufId, XStart, YStart, XEnd, YEnd	Draw a rectangle.
MgraRectFill()	GraphContId, DestImageBufId, XStart, YStart, XEnd, YEnd	Draw a filled rectangle.
MgraText()	GraphContId, DestImageBufId, XStart, YStart, String	Write text.

Image processing module

Used to perform filtering, morphological, point-to-point, segmentation, and statistical operations on an image. This module also includes geometric, color space, and domain transforms, as well as other image processing primitives.

Commands	Command parameters	Description
MimAllocResult()	SystemId, NbEntries, ResultType, ImResultIdPtr	Allocate an image processing result buffer.
MimArith()	Src1ImageBufId, Src2ImageBufId, DestImageBufId, Operation	Perform a point-to-point arithmetic operation.
MimArithMultiple()	Src1ImageBufId, Src2ImageBufId, Src3ImageBufId, Src4ImageBufId, Src5ImageBufId, DestImageBufId, Operation, OperationFlag	Perform a point-to-point arithmetic operation using multiple source images.
MimBinarize()	SrcImageBufId, DestImageBufId, Condition, CondLow, CondHigh	Perform a point-to-point binary thresholding operation.
MimClip()	SrcImageBufId, DestImageBufId, Condition, CondLow, CondHigh, WriteLow, WriteHigh	Perform a point-to-point clipping operation.
MimClose()	SrcImageBufId, DestImageBufId, NbIteration, ProcMode	Perform a binary or grayscale closing-type morphological operation.
MimConnectMap()	SrcImageBufId, DestImageBufId, LutBufId	Perform a 3 by 3 binary connectivity mapping.
MimConvert()	SrcImageId, DestImageId, ConversionType	Perform a color conversion.
MimConvolve()	SrcImageBufId, DestImageBufId, KernelBufId	Perform a general convolution operation.
MimCountDifference()	Src1ImageBufId, Src2ImageBufId, ImResultId	Count the number of pixels that differ in each image.
MimDeinterlace()*	ContextId, SrcImageArrayPtr, DstImageArrayPtr, SrcImageCount, DstImageCount, ControlFlag	Produce a sequence of deinterlaced images from a sequence of images acquired from an interlaced camera.
MimDilate()	SrcImageBufId, DestImageBufId, NbIteration, ProcMode	Perform a binary or grayscale dilation-type morphological operation.
MimDistance()	ScrImageBufId, DestImageBufId, DistanceTransform	Perform a distance transformation.
MimEdgeDetect()	SrcImageBufId, DestIntensityImageBufId, DestAngleImageBufId, KernelId, ControlFlag, Threshold	Perform a specific edge detection operation and produce a gradient intensity and/or gradient angle image.
MimErode()	SrcImageBufId, DestImageBufId, NbIteration, ProcMode	Perform an erosion-type morphological operation.
MimFindExtreme()	SrcImageBufId, ExtremeImResultId, ExtremeType	Find an image buffer's extremes (minimum and/or maximum pixel values)
MimFlip()	ScrImageId, DestImageId, Operation, OpFlag	Perform a horizontal or vertical image-flipping rotation.
MimFree()	ImResultId	Free an image processing result buffer.
MimGetResult()	ImResultId, ResultType, UserArrayPtr	Get values from an image processing result buffer.
MimGetResult1d()	ImResultId, OffEntry, NbEntries, ResultType, UserArrayPtr	Get values from a 1D region of an image processing result buffer.
MimHistogram()	SrcImageBufId, HistImResultId	Generate the intensity histogram of an image buffer.

* Available as of Processing Pack 1.

Image processing module (continued)

Commands	Command parameters	Description
MimHistogramEqualize()	SrcImageBufId, DestImageBufId, Method, Alpha, Min, Max	Perform a histogram equalization of an image.
MimInquire()	BufId, InquireType, UserVarPtr	Inquire about an image processing result buffer parameter setting.
MimLabel()	SrcImageBufId, DestImageBufId, ProcMode	Label objects in an image buffer.
MimLocateEvent()	SrcImageBufId, EventImResultId, Condition, CondLow, CondHigh	Find pixel coordinates or values that satisfies a specified condition.
MimLutMap()	SrcImageBufId, DestImageBufId, LutBufId	Perform a point-to-point LUT mapping operation.
MimMorphic()	SrcImageBufId, DestImageBufId, StructElemBufId, Operation, NBIteration, ProcMode	Perform a morphological transformation using a user-defined kernel.
MimOpen()	SrcImageBufId, DestImageBufId, NbIteration, ProcMode	Perform a binary or grayscale opening-type morphological operation.
MimPolarTransform()	SrcImageBufId, DestImageBufId, CenterPosX, CenterPosY, StartRadius, EndRadius, StartAngle, EndAngle, OperationMode, InterpolationMode, DestSizeXPtr, DestSizeYPtr	Perform a polar-to-rectangular or rectangular-to-polar transforms.
MimProject()	SrcImageBufId, ProjImResultId, ProjAngle	Project a 2D image into 1D.
MimRank()	SrcImageBufId, DestImageBufId, StructElemBufId, Rank, ProcMode	Perform a rank filter on the pixels in an image.
MimResize()	SrcImageBufId, DestImageBufId, ScaleFactorX, ScaleFactorY, InterpolationMode	Resize an image.
MimRotate()	SrcImageBufId, DestImageBufId, Angle, SrcCenX, SrcCenY, DstCenX, DstCenY, InterpolationMode	Rotate an image.
MimShift()	SrcImageBufId, DestImageBufId, BitsToShift	Perform a point-to-point bit shift.
MimStat()	SrcImageId, StatResultId, StatType, Condition, CondLow, CondHigh,	Calculate a variety of statistics on the source image.
MimThick()	SrcImageBufId, DestImageBufId NbIteration, ProcMode	Perform a binary or grayscale thickening operation on an image.
MimThin()	SrcImageBufId, DestImageBufId NbIteration, ProcMode	Perform a binary or grayscale thinning operation on an image.
MimTransform()	SrcImageRBufId, SrcImageIBufId, DestImageRBufId, DestImageIBufId TransformType, ControlFlag	Perform a Fast Fourier transform (FFT) or a Discrete Cosine transform (DCT).
MimTranslate()	SrcImageBufId, DestImageBufId, XDisplacement, YDisplacement, InterpolationMode	Translate an image in X and/or Y displacement.
MimWarp()	SrcImageId, DestImageId, WarpParam1Id, WarpParam2Id, OperationMode, InterpolationType	Perform a warping.
MimWatershed	SrcImageId, MarkerImageId, DestImageId, MinimumVariation, ControlFlag	Perform a watershed transformation.
MimZoneOfInfluence()	SrcImageBufId, DestImageBufId, OperationFlag	Perform a zone of influence detection.

Measurement module

Used to locate and measure edges or stripes within an image. Also used to take measurements between points, edges, or stripes. This module includes functions to save or restore markers (i.e., points, edges, or stripes).

Commands	Command parameters	Description
MmeasAllocContext()	SystemId, ControlFlag, ContextIdPtr	Allocate a measurement context.
MmeasAllocMarker()	SystemId, MarkerType, ControlFlag, MarkerIdPtr	Allocate a measurement marker.
MmeasAllocResult()	SystemId, ResultType, MeasResultIdPtr	Allocate a measurement result buffer.
MmeasCalculate()	ContextId, Marker1Id, Marker2Id, MeasResultId, MeasurementList	Calculate measurements between two markers.
MmeasControl()	ContextId, ControlType, Value	Control a measurement parameter setting.
MmeasDraw()	GraphContId, MarkerOrResultId, DestImageId, Operation, Index, ControlFlag	Draw features of specific markers or result occurrences in an image buffer.
MmeasFindMarker()	ContextId, ImageBufId, MarkerId, MeasurementList	Find a marker in an image and take the specified measurements.
MmeasFree()	MeasId	Free a measurement context, marker, or result buffer.
MmeasGetResult()	MarkerOrMeasResultId, ResultType, FirstResultArrayPtr, SecondResultArrayPtr	Get the results of measurements taken.
MmeasGetResultSingle()	MarkerOrMeasResultId, ResultType, FirstResultArrayPtr, SecondResultArrayPtr, ResultIndex	Get a single result from a multiple marker or its result buffer.
MmeasInquire()	MeasId, InquireType, FirstValuePtr, SecondValuePtr	Inquire about a measurement context, marker, or result buffer.
MmeasRestoreMarker()	FileName, SystemId, ControlFlag, MarkerIdPtr	Restore a marker from disk.
MmeasSaveMarker()	FileName, MarkerId, ControlFlag	Save a marker to disk.
MmeasSetMarker()	MarkerId, CharacteristicToSet, FirstValue, SecondValue	Set a marker characteristic parameter.

OCR module

Template-based character recognition module. This module includes control of character font definition, as well as font archiving and retrieving.

Commands	Command parameters	Description
MocrAllocFont()	SystemId, FontType, CharNumber, CharBoxSizeX, CharBoxSizeY, CharOffsetX, CharOffsetY, CharSizeX CharSizeY, CharThickness, StringLength, InitFlag, FontIdPtr	Allocate an OCR font buffer.
MocrAllocResult()	SystemId, InitFlag, OcrResultIdPtr	Allocate an OCR result buffer.
MocrCalibrateFont()	ImageBufId, FontId, String, TargetCharSizeXMin, TargetCharSizeXMax, TargetCharSizeXStep, TargetCharSizeYMin, TargetCharSizeYMax, TargetCharSizeYStep, Operation	Calibrate font character size to match a sample image.
MocrControl()	FontId, ControlToSet, Value	Control an OCR parameter setting.
MocrCopyFont()	ImageBufId, FontId, Operation, CharListString	Copy a font character to or from an image buffer.
MocrFree()	FontIdOrResultId	Free an OCR font or result buffer.
MocrGetResult()	OcrResultId, ResultToGet, ResultPtr	Read results from an OCR result buffer.
MocrHookFunction()	FontId, HookType, HookHandlerPtr, UserDataPtr	Hook a function to an event.
MocrImportFont()	FileName, FileFormat, Operation, CharListString, FontId	Import font data from file on disk.
MocrInquire()	FontId, InquireItem,UserVarPtr	Inquire about font character information.
MocrModifyFont()	FontId, Operation, ControlValue	Invert or resize a font to match the target image characters.
MocrPreprocess()	FontId, ControlFlag	Preprocess an OCR font context.
MocrReadString()	ImageBufId, FontId, OcrResultId	Read an unknown string from an image.
MocrRestoreFont()	FileName, Operation, SystemId, FontIdPtr	Restore a font from disk.
MocrSaveFont()	FileName, Operation, FontId	Save an existing font to disk.
MocrSetConstraint()	FontId, CharPos, CharPosType, CharValidString	Set character position constraints.
MocrVerifyString()	ImageBufId, FontId, String, OcrResultId	Verify a known string in an image.

Pattern matching module

Used to locate patterns in an image using normalized grayscale correlation (NGC). This module includes functions to define a pattern, control search strategy, and save and restore a pattern.

Commands	Command parameters	Description
MpatAllocAutoModel()	SystemId, SrcImageBufId, SizeX, SizeY, PosUncertaintyX, PostUncertaintyY, ModelType, Mode, ModelIdPtr	Automatically allocate unique pattern matching models of the specified type, from a source image.
MpatAllocModel()	SystemId, SrcImageBufId, OffX, OffY, SizeX, SizeY, ModelType, ModelIdPtr	Allocate a pattern matching model from a source image.
MpatAllocResult()	SystemId, NbEntries, PatResultIdPtr	Allocate a pattern matching result buffer.
MpatAllocRotatedModel()	SystemId, SrcModelId, Angle, InterpolationMode, ModelType, NewModelIdPtr	Rotate a pattern matching model.
MpatCopy()	ModelId, DestImageBufId, CopyMode	Copy a pattern matching model to an image buffer.
MpatDraw()	GraphContId, ModelOrResultId, DestImageId, Operation, Index, ControlFlag	Draw features of a specific model or result occurrences in an image buffer.
MpatFindModel()	ImageBufId, ModelId, PatResultId	Find a pattern matching model in the target image buffer.
MpatFindMultipleModel()	ImageBufId, ModelIdLst, PatResultIdst, NumModels, ExpFlag	Find multiple pattern matching models in the target image buffer.
MpatFree()	PatId	Free a pattern matching model or a result buffer.
MpatGetNumber()	PatResultId, CountPtr	Get the number of model occurrences in the target image.
MpatGetResult()	PatResultId, ResultType, UserArrayPtr	Get the pattern matching result values.
MpatInquire()	PatId, ParamToInquire, UserVarPtr	Inquire about the pattern matching model or the result buffer parameter setting.
MpatPreprocModel()	TypicalImageBufId, ModelId, Mode	Preprocess a pattern matching model.
MpatRead()	SystemId, FileHandle, ModelIdPtr	Read a pattern matching model from an open file.
MpatRestore()	SystemId, FileName, ModelIdPtr	Restore a pattern matching model from disk.
MpatSave()	FileName, ModelId	Save a pattern matching model to disk.
MpatSetAcceptance()	ModelId, AcceptanceThreshold	Set the acceptance level of a model.
MpatSetAccuracy()	ModelId, Accuracy	Set the positional accuracy of a model.
MpatSetAngle()	ModelId, ControlType, ControlValue	Set the angular search parameters of a model.
MpatSetCenter()	ModelId, OffX, OffY	Set the reference position of a model.
MpatSetCertainty()	ModelId, CertaintyThreshold	Set the certainty level of a model.
MpatSetDontCare()	ModelId, ImageBufId, OffX, OffY, Value	Set the “don’t care” pixels in a model.
MpatSetNumber()	ModelId, NbOccurences	Set the expected number of occurrences of a model.
MpatSetPosition()	ModelId, OffX, OffY, SizeX, SizeY	Set the search region of a model.
MpatSetSearchParameter()	PatId, Parameter, Value	Set the internal search parameters of a model.
MpatSetSpeed()	ModelId, SpeedFactor	Set search speed of a model.
MpatWrite()	FileHandle, ModelId	Write a pattern matching model to an open file.

String Reader module

Feature-based character recognition module. This module supports multiple user-defined grammar rules and multi-font definition in a single context.

Commands	Command parameters	Description
MstrAlloc()	SystemId, ContextType, ControlFlag, ObjectIdPtr	Allocates a String Reader context.
MstrAllocResult()	SystemId, ControlFlag, ObjectIdPtr	Allocate a String Reader result buffer.
MstrControl()	ContextOrResultID, Index, ControlType ControlValue	Control a String Reader context, a specific string model, a specific font, or a String Reader result buffer.
MstrDraw()	GraphContId, ContextOrResultID, DestImageId, Operation, Index, CharList ControlFlag	Draw specific features of the String Reader context or String Reader results.
MstrEditFont()	ContextId, FontIndex, Operation, OperationMode, Param1, Param2, Param3	Edit a specified font.
MstrFree()	ObjectId	Free a String Reader context or a String Reader result buffer.
MstrGetResult()	ResultId, Index, ResultType, ResultArrayPtr	Get the specified type of result(s) from a String Reader result buffer.
MstrInquire()	ContextOrResultId, Index, InquireType UserVarPtr,	Inquire information about a specified String Reader context, string model, font, or result buffer.
MstrPreprocess()	ContextId, ControlFlag	Preprocess a String Reader context.
MstrRead()	ContextId, TargetImageId, ResultId	Read strings from a target image.
MstrRestore()	Filename, SystemId, ControlFlag, ContextIdPtr	Restore a String Reader context from disk.
MstrSave()	FileName, ContextId, ControlFlag	Save a String Reader context to a file.
MstrSetConstraint()	ContextId, StringIndex, CharPos ConstraintType, CharList	Set character constraints.
MstrStream()	MemPtrOrFileName, SystemId, Operation, StreamType, Version ControlFlag, ObjectIdPtr, SizeByteVarPtr	Load, restore, or save a String Reader context from/to a file or a memory.

Thread module

Used for the allocation of MIL thread contexts and synchronization events. This module allows control over the created MIL thread contexts and events, inquire about various settings, and synchronize execution of multiple threads.

Commands	Command parameters	Description
MthrAlloc()	SystemId, ObjectType, ControlFlag ThreadFctPtr, UserPtr, ThreadOrEventId	Allocate a MIL thread context or event.
MthrControl()	ThreadOrEventId, ControlType, ControlValue	Control MIL thread context or MIL event settings.
MthrFree()	ThreadOrEventId	Free a MIL thread context or event.
MthrInquire()	ThreadOrEventId, InquireType, InquireValue	Inquire about a MIL thread context or event setting.
MthrWait()	ThreadOrEventId, WaitOption, State	Perform a wait operation on a MIL thread or event.