

# Type0003 Module 使用方法について

第1版

2009年4月13日

## 1 概要

MAID3.0 規約、Type0003MAID 規約には述べられていない、Type0003 Module を使用するにあたって注意すべき事柄について述べる。いくつかは現在の Module における制限である。

## 2 サポートするカメラ

本 SDK に添付の Type0003 Module でサポートするカメラは、D90、D5000 となる。

また、コントロール可能なカメラは1台のみで、複数台のカメラコントロールは出来ない。

## 3 動作環境

OS タイプ	Version
Windows	Windows XP Home Edition (SP3) / Professional (SP3) Windows Vista SP1 32bit 版の各エディション (Home Basic / Home Premium / Business / Enterprise / Ultimate)
Macintosh	MacOS X 10.4.11 MacOS X 10.5.5

## 4 Capability 関連

Source object を open した後、Client は必ず一度、各 Capability の現在値を取得する必要がある。(値を設定する前に、毎回現在値を取得する必要は無い) 一度も現在値を取得せずに、kNkMAIDCommand\_CapSet により、値の設定を実行した場合、Client の Set した値がカメラに正しく設定されない場合がある。

### 4.1 kNkMAIDCapability\_ProgressProc

Module は進行情報を、MAIDProgress 関数の呼び出しにより Client に対して通知する。Module はどの程度処理が終了したのかを判断できない場合、ulTotal=0 かつ ulDone≠0 として呼び出す。この処理が終了した場合には、ulDone=ulTotal として呼び出す。

### 4.2 kNkMAIDCapability\_EventProc

MAID3.1 規約では、MAIDEvent 関数を設定するかどうかは、Client の自由とされているが、現在の Module は、常に Client が設定することを前提としている。このため Client が MAIDEvent 関数を設定しない場合には、いくつかの制限が発生する。

- 1)kNkMAIDCommand\_EnumChildren が使用できない。
- 2)レンズ交換、装置の電源 ON-OFF 等に対応することができない。
- 3)Capability の値が変化しても通知されないため、Client は Capability の値を常に監視しつづけなければならない。

### 4.3 kNkMAIDCapability\_Children

ある Object が持つ子オブジェクトを列挙するために使用される。同様の機能として、kNkMAIDCommand\_EnumChildren を使用することもできる。EventProc に MAIDEvent 関数を設定しない場合には、子オブジェクトを列挙するためには kNkMAIDCapability\_Children を使用しなければならない。

#### 4.4 kNkMAIDCapability\_PictureControlData

下記のピクチャコントロールデータの各項目の設定内容により、カメラはピクチャコントロールデータの設定値を使用するか、または、カメラ自身が内部的に決めた値を使用するかを決定する。

##### 1) QuickAdjustFlag (カラー)

有効(1)の場合、カメラは QuickAdjust のみを使用する。

無効(0)の場合、Saturation、Hue、Sharpening、Contrast、Brightness、CustomCurveFlag、CustomCurveData を使用し、QuickAdjust は使用しない。

##### 2) CustomCurveFlag

カスタムカーブ使用(1)の場合、カメラは Contrast、Brightness を使用しない。

##### 3) Toning (モノクロ)

B&W(0)の場合、カメラは ToningDensity を使用しない。

##### 4) Contrast、Brightness、CustomCurveFlag、CustomCurveData

kNkMAIDCapability\_Active\_D\_Lighting が”しない”(3)以外の場合、カメラは Contrast、Brightness、CustomCurveFlag、CustomCurveData を使用しない。

#### 4.5 kNkMAIDCapability\_DeleteDramImage

DRAM 内に保存された撮影画像を削除するタイミングは、下記の 2 つのタイミングに限られる。

下記のタイミング以外での DRAM 画像削除はサポートされない。

1. SourceObject に対する kNkMAIDEvent\_AddChild イベントを受信する前
2. ImageObject に対する kNkMAIDCapability\_Acquire を発行した後で、kNkMAIDCommand\_Close を発行する前

1 の場合のコマンド発行手順例を示す。

No	Command / Capability / Event	Object Type
1	kNkMAIDCapability_Capture	Source
2	kNkMAIDCommand_Async	Source
3	kNkMAIDEvent_AddPreviewImage	Source
4	kNkMAIDCapability_CurrentPreviewID	Source
5	kNkMAIDCapability_DeleteDramImage	Source

2 の場合のコマンド発行手順例を示す。

No	Command / Capability / Event	Object Type
1	kNkMAIDCapability_Capture	Source
2	kNkMAIDCapability_Children	Source
3	kNkMAIDCommand_Open	Item
4	kNkMAIDCapability_Children	Item
5	kNkMAIDCommand_Open	Image
6	kNkMAIDCapability_DataProc (Set)	Image
7	kNkMAIDCapability_Acquire	Image
8	kNkMAIDCommand_Async	Image
9	kNkMAIDCommand_Abort	Image

10	kNkMAIDCapability_CurrentPreviewID	Source
11	kNkMAIDCapability_DeleteDramImage	Source
12	kNkMAIDCapability_DataProc (Reset)	Image
13	kNkMAIDCommand_Close	Image
14	kNkMAIDCommand_Close	Item

kNkMAIDCapability\_ProgressProc でコールバック関数を設定している場合、処理終了時にはコールバック関数のパラメータが「ulDone==ulTotal」、または「ulDone==ulTotal==0」に Set されることにより、処理完了が通知される仕様となっているが、kNkMAIDCommand\_Abort により処理を中断した場合、コールバック関数に対する処理完了は通知されない。

#### 4.6 kNkMAIDCapability\_GetPreviewImageNormal

画質 Normal のプレビューデータは、取得出来ない場合がある。画像 Normal のプレビューデータは、主画像の取り込みが終了した時点でカメラ内部から削除される仕様となっており、現状の Module は、クライアントの要求有無に関係なく、カメラから主画像の生成完了通知を受信した時点で、直ぐに主画像の取得を開始する仕様となっており、Module が主画像の取得が完了した時点で画質 Normal のプレビューデータはカメラ内部により削除される。よって、画質 Normal のプレビューデータの取得は 100% 保障される機能ではない。

但し、Normal プレビューデータ取得に一度でも成功すると、モジュール内部でキャッシュが生成されるため、Item についての kNkMAIDEvent\_AddChild イベントを受信以降も、Item を close するまでの間、何度でも取得可能である。Low プレビューに関しては、Module は主画像取得前に読み込んでキャッシュを作成するため、Item を close するまでの間、何度でも取得可能である。

### 5 イメージおよびサムネイルデータ

イメージデータは、MAID Data Delivery Function によりファイルデータとして受け渡される。(MAID3.DOC 5.27 File Data Delivery Structure および 10.3 MAID Data Delivery Function 参照)

サムネイルデータは、ヘッダ無しの RAW データでありファイルデータではない。並び方は、点順次 (R G B R G B ・ ・ ・) で、画素並びは左から右、上から下の順である。元画像のフォーマットに関わらず同一形式であり、サイズは幅 160×高さ 120pixel で固定である。

### 6 カメラとの接続・切断

Client は、Module オブジェクトに対して定期的に kNkMAIDCommand\_Async を発行していれば、カメラが接続された時 AddChild Event によりそれを知ることができる。カメラとの接続が断たれた場合、オープン中の Source オブジェクトに対して RemoveChild Event が発行される。

### 7 Object のオープン

Module、Source および Item Object は、同じオブジェクト型に属するオブジェクトを同時に 2 つ以上 Open することは出来ない。同時に Open することが出来るのは、1 つのオブジェクト型につき 1 つのみとなる。(例えば、ID の異なる Source Object が 2 つ存在しても、Open 可能なのはどちらか 1 つとなる)

但し、Image および Thumbnail Object については、同じ kNkMAIDObjectType\_DataObj に属するが、同じ Item Object から 2 つを同時に Open することが可能である。

## 8 ライブビュー実行中の制限について

ライブビューの実行中は、多くの Capability が使用禁止となる。Type0003 MAID 規約内で、ライブビュー中の使用が可能であるとの明確な記述がない限り、ライブビュー中の Capability 実行は禁止とする。

但し、ライブビュー実行により使用禁止状態となった Capability の ulVisibility、ulOperations は、基本的に変更されず、そのままの状態となる仕様とする。（Capability\_AFCapture、Capability\_PreCapture など、一部例外もあり）

ライブビュー中に実行可能な Capability は以下の通り。

kNkMAIDCapability_ContrastAF
kNkMAIDCapability_MFDriveStep
kNkMAIDCapability_MFDrive
kNkMAIDCapability_ContrastAFArea
kNkMAIDCapability_LiveViewStatus
kNkMAIDCapability_LiveViewProhibit
kNkMAIDCapability_LiveViewImageZoomRate
kNkMAIDCapability_Capture
kNkMAIDCapability_GetLiveViewImage

## 9 構造体のアライメント

以下に構造体のアライメントを示す。MAID3.H 内にアライメントが 4byte であるという記述があるが、実際にはプラットフォームによって、異なった値になっている。

プラットフォーム	アライメント
Windows 32bit	2 byte
Macintosh	4 byte

## 10 Macintosh 版モジュールについて

Macintosh 版の Type0003 ジュールは、Universal Binary となる。