

CONTENTS

ビジュアルヘルプ – MMI (Micro-Manager Interface)	2
概要	2
セットアップ	3
Micro-Manager のインストール.....	4
Micro-Manager、ImageJ、Java.....	5
手動で Micro-Manager をインストール	5
予備的なメモ.....	7
Igor ウェーブでの画像の保存と表示に関する注意.....	8
MMI の開始手順.....	9
最初の例.....	10
画像のストリーミング	11
画像とストリームの保存	12
次のステップ	14
インターフェイスのサマリ.....	14
管理インターフェイス.....	14
Micro-Manager コアインターフェイス.....	16
インターフェイスのメソッド	19
Micro-Manager のメソッド	40
初期化とセットアップ	40
ログ収集とログ管理	45
デバイスのリスト.....	49

ビジュアルヘルプ – MMI (Micro-Manager Interface)

概要

MMI (Micro-Manager Interface) は、Igor Pro® v.10 用の 64 ビット拡張 (XOP) で、Igor と Micro-Manager™ v.2.0 を連携させます。

Micro-Manager <<https://micro-manager.org/>> は、光学顕微鏡の自動化のためのオープンソースのソフトウェアパッケージで、もともと画像アプリケーション ImageJ (<<https://imagej.net/ij/>>) と連携して動作するように開発されました。

Micro-Manager は、科学用カメラ、光源、シャッター、フィルターホイール、XY 軸および Z 軸ステージ、モーター駆動式顕微鏡など、多様なデバイスをコントロール可能です。

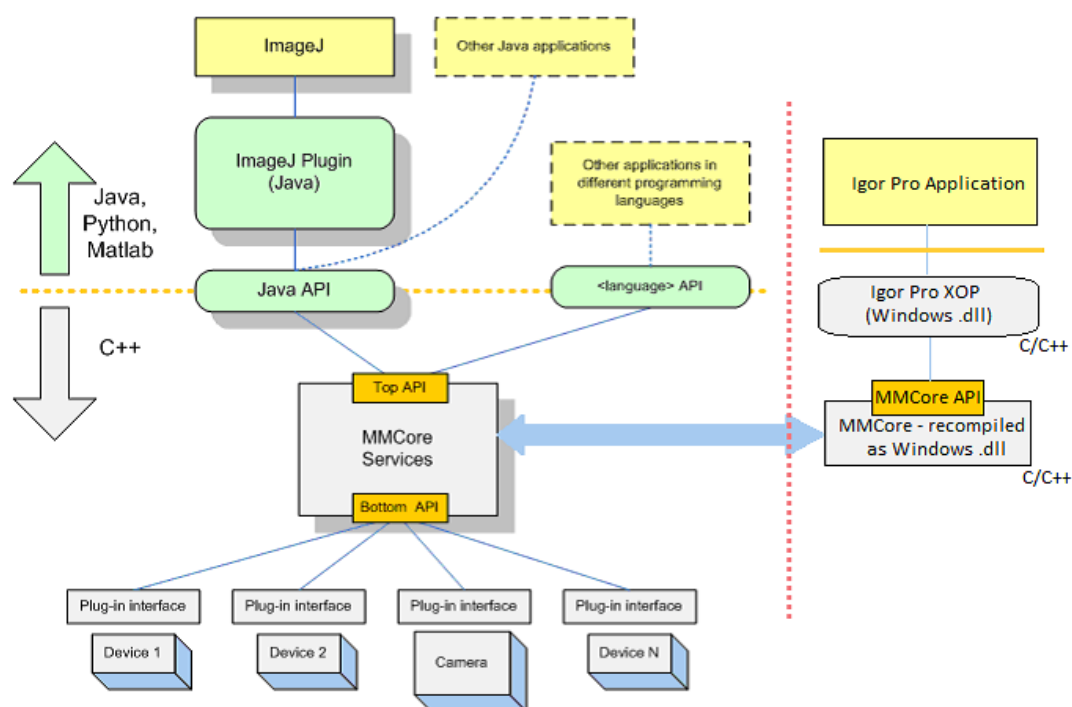
Igor Pro、MMI、Micro-Manager を一緒に使うことで、Igor プロシージャはこれらのデバイスをコントロールして、データの取得、保存、分析を行うことができます。

下の図は、Micro-Manager ソフトウェアのアーキテクチャと、Igor Pro と Micro-Manager 間の MMI インターフェイスをまとめたものです。

赤い点線の左側に表示されているモジュールは、Micro-Manager の通常の構造を示しています。

赤い点線の右側のモジュールは、MMI インターフェイスと Igor Pro を示しています。

(図は、<https://micro-manager.org/media/Block_diagram.gif>から改変したものです)



いくつかの単純な例を除き、MMI は Micro-Manager への GUI (グラフィカルユーザーインターフェイス) を提供していません。

MMI は、Igor で独自のプロシージャを記述するユーザー向けに設計されています。

このアプローチの利点は、ユーザーが GUI を使わずに実行することも、自分のニーズに最も合った GUI を構築することもできることです。

MMI は、2023 年 9 月 9 日にビルドされた Micro-Manager の特定のビルド (バージョン 2.0.3) と組み合わせて動作するように設計されています。

複数の異なるバージョンの Micro-Manager が、互いに干渉することなく、1 つのコンピューター上に共存可能です。

従って、コンピューターにすでに別のバージョンの Micro-Manager がインストールされている場合でも、MMI に必

要とされるバージョンをインストールし、必要に応じて環境変数を設定する方が良いでしょう（詳細は「Micro-Manager のインストール」を参照）。

複数のバージョンの Micro-Manger を「並列インストール」しても、問題が発生しないはずです。

MMI のほとんどのコマンドと機能は、Micro-Manager の MMCore クラスによって公開されている API の機能の構成に従っています。

詳細は <https://micro-manager.org/apidoc/MMCore/latest/class_c_m_m_core.html> を参照してください。

MMI は通常、値を返さない Micro-Manager の機能（つまり、C 言語の「void」戻り値型を使う機能）を呼び出すために、Igor Pro コマンドを使います。

MMI は、明示的に値を返す Micro-Manager の機能呼び出しを行うために、Igor Pro 機能を使っています。

Micro-Manager の関数が数値、ブール値、または文字列の結果を返す場合、MMI はその結果を数値または文字列値として Igor に返します。

Micro-Manager の関数が結果の配列またはベクトルを返す場合、MMI は通常、結果を Igor ウェーブとして返します。

MMI と Micro-Manager はどちらも進行中のプロジェクトであることに注意してください。

そのため、Micro-Manager が提供する機能は今後変更される可能性があります。

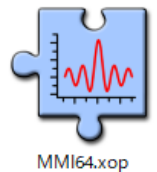
さらに、MMI は、Micro-Manager の CMMCore クラスが露出しているすべての公開関数へのアクセスを提供していません。

CMMCore API のパブリック関数がこの MMI ドキュメントのリストに表示されていない場合、その関数はまだサポートされていないものとみなすことができます。

セットアップ

MMI を実装するファイルは、当然ながら MMI64.xop と呼ばれています。

このファイルは、依存する他のファイルとともに、Igor 10 のサブディレクトリ「.¥More Extensions (64-bit) ¥Data Acquisition¥MMI」にインストールされます。



MMI をアクティブにするには、Igor の「.¥Igor Extensions (64-bit)」サブディレクトリにある MMI64.xop ファイルへの Windows ショートカットを作成することをお勧めします。

これを行う最も簡単な方法の一つは、Windows エクスプローラーを開き、ディレクトリ「.¥More Extensions (64-bit) ¥Data Acquisition¥MMI」をブラウズします。

ファイル「MMI64.xop」にカーソルを合わせ、右クリックボタンを押したままドラッグし、ファイルのアイコンを Windows デスクトップにドラッグアンドドロップします。

右クリックボタンを離すと、Windows はポップアップメニューを表示します。

そのメニューから「ショートカットをここに作成」という項目をクリックしてください。

Windows はショートカットを作成し、デスクトップにショートカットのアイコンが表示されます。

次に、Windows エクスプローラーを使って、「.¥Igor Extensions (64-bit)」ディレクトリに移動します。

作成したデスクトップショートカットをクリックし、「Igor Extensions (64-bit)」ディレクトリにドラッグ&ドロップします。

これらの手順が完了したら、Igor Pro を起動（または再起動）し、次のコマンドを入力してください。

```
print MMI_GetXOPversion()
```

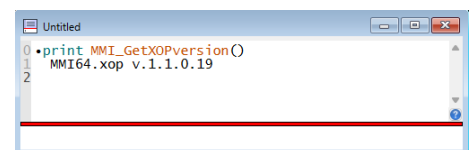
次のような応答が表示されるはずです。

```
MMI64.xop v.1.1.0.19
```

これで、MMI XOP がアクティブ化されました（つまり、Igor が XOP をロードした）が、必要なバージョンの Micro-Manager にはまだリンクされていません。

MMI は、MMI_GetXOPversion() 以外のコマンドを要求されると、Micro-Manager へのインターフェイスを起動しようとします。

MMI が選択した Micro-Manager のバージョンを見つけた場合、そのインターフェイスファイルが所定の場所に存



在することを確認し、必要に応じてそのインターフェイスファイルをその場所にコピーします。
インターフェイスファイルが配置されると、MMI はインターフェイスを起動し、内部的な起動を完了します。
この起動中に、MMI が Micro-Manager を検出できなかった場合、またはインストールされた Micro-Manager に問題が見つかった場合、Micro-Manager のダウンロードまたはインストールを促すメッセージが表示されます（次のセクションと MMI_DownloadMM と MMI_InstallMM を参照してください）。

Micro-Manager のインストール

MMI の Micro-Manager のダウンロードとインストールには、Windows の管理者権限が必要な処理（例：環境変数の作成、Micro-Manager インストーラーの実行）を実行します。

これらのタスクを実行する前に、Windows はこれらのタスクを実行する許可を求める可能性があります。
Micro-Manger のインストールを簡素化するため、Igor Pro を管理者権限で実行することをおすすめします。
MMI がこれらのタスクを実行することを承認しない場合は、以下の「手動で Micro-Manager をインストール」セクションの手順に従ってください。

MMI は、必要なバージョンの Micro-Manager が正しくインストールされているかどうかを判断するために、複数の Windows 環境変数を使います。

このプログラムは、Micro-Manager がインストールされているディレクトリのパスを格納する変数 MM_BIN に依存しています。

つまり、Micro-Manager の実行ファイル（.exe ファイル）とライブラリ（.dll ファイル）がインストールされているディレクトリを指します。

MM_BIN が存在しない場合、MMI は Micro-Manager がまだインストールされていないものと判断し、Micro-Manager インストーラーのパスを含む環境変数 MM_INSTALLER を検索します。

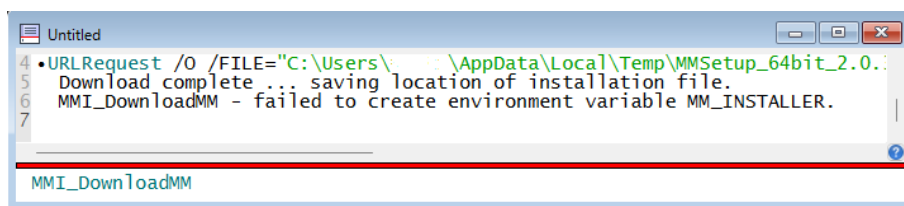
MM_INSTALLER 変数も存在しない場合、MMI は MMI_DownloadMM を実行するように促します。

これは、インターネットに接続可能なコンピュータ上で、Micro-Manager ウェブサイトから必要なインストーラーをダウンロードし、コンピュータに保存します。

MMI は次に MM_INSTALLER を作成し、その中にダウンロードしたインストーラーのパスを保存します。

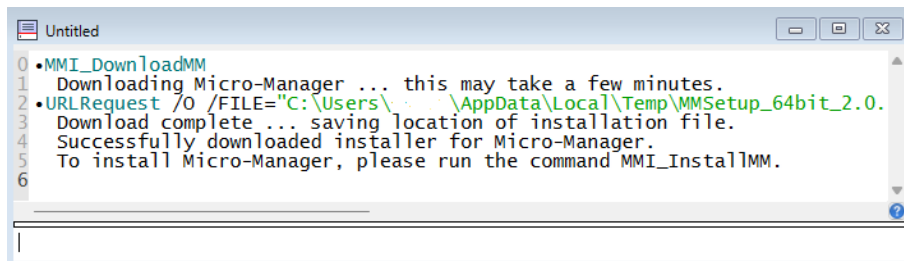
Igor Pro のコマンドウィンドウで次を実行します。

MMI_DownloadMM



```
Untitled
4 •URLRequest /O /FILE="C:\Users\... \AppData\Local\Temp\MMSetup_64bit_2.0.
5 Download complete ... saving location of installation file.
6 MMI_DownloadMM - failed to create environment variable MM_INSTALLER.
7
```

（MM_INSTALLER 環境変数が作成できないとき、には左のようになります。Igor Pro を「管理者として実行」すると、環境変数が作成できます。）



```
Untitled
0 •MMI_DownloadMM
1 Downloading Micro-Manager ... this may take a few minutes.
2 •URLRequest /O /FILE="C:\Users\... \AppData\Local\Temp\MMSetup_64bit_2.0.
3 Download complete ... saving location of installation file.
4 Successfully downloaded installer for Micro-Manager.
5 To install Micro-Manager, please run the command MMI_InstallMM.
6
```

（成功すると、次に MMI_InstallMM コマンドを実行するように表示されます。）

MM_BIN は存在しないが MM_INSTALLER が存在する場合、MMI は MMI_InstallMM を実行するように促します。

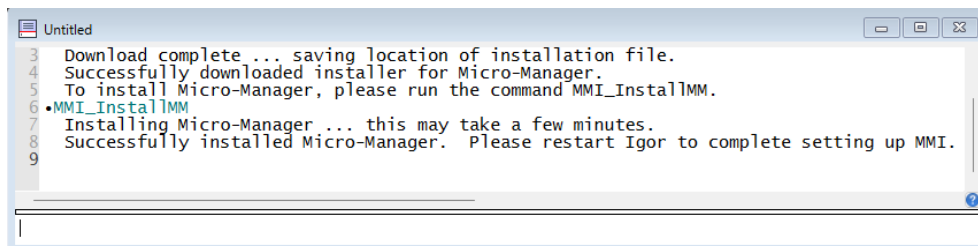
MMI_InstallMM

この MMI_InstallMM は、ここまでにダウンロードされた Micro-Manager のインストーラーを実行します。
この操作は、MM_BIN を作成し、その中に Micro-Manger がインストールされたディレクトリのパスを格納します。

一部の Windows セキュリティソフトウェア（特に Windows Defender）は、Micro-Manager の認識されていないインストーラーを実行しようとするのを防ぐ場合があるため、注意してください。

Micro-Manager をインストールする場合、この保護機能を無効化し、インストーラーを実行する必要があります。Windows Defender の場合、これは「詳細情報」というテキストをクリックし、表示されるダイアログで「Run Anyway」というラベルのついたボタンをクリックすることを意味します。

Micro-Manager のインストールが正常に完了すると、XOP の初期化コードにより、必要なインターフェイスファイル MMCoreDyn.dll が MM_BIN ディレクトリにコピーされ、MM_BIN ディレクトリがシステム PATH 変数に追加されます。



インストールと設定が完了すると、「Successfully installed Micro-Manager. Please restart Igor to complete setting up MMI.」というメッセージが表示されます。

これで、MMI のインストールは完了し、Igor を再起動すると、使用可能になります。

Micro-Manager、ImageJ、Java

Micro-Manager の通常のインストールには、Java ベースのイメージングアプリケーションである ImageJ.exe が含まれます。

Igor Pro および MMI は Java および ImageJ から独立しているため、ImageJ および Micro-Manager の Java コンポーネントは、必要に応じて残すか削除するかを選択できます。

MMI_InstallMM を呼び出して Micro-Manager をインストールする場合、インストール時に Micro-Manager の Java ベースのコンポーネントを自動的に削除するには /RJ フラグを使用できます。

Micro-Manager を手動でインストールする場合（下記参照）、必要に応じて Java コンポーネントを手動で削除する必要があります。

ImageJ と Java をインストールしたままにしておけば、他の Windows プログラムと同様に、ImageJ 経由で Micro-Manager を実行するオプションを維持できます。

ImageJ から Micro-Manager を実行すると、Igor Pro および MMI を完全にバイパスして、標準のユーザーインターフェイスが表示されます。

この経路を使うと、Igor および MMI とは独立して、Micro-Manager による機器のコントロールをテストすることができます。

これは、Igor のコードを開発する時の貴重なデバッグ支援ツールとなります。

Micro-Manager のネイティブユーザーインターフェイスについて詳しくは、<https://micro-manager.org/Micro-Manager_Project_Overview> を参照してください。

手動で Micro-Manager をインストール（Igor からインストールしない場合）

何らかの理由で、MMI の組み込みコマンドを使わずに、Igor 用 Micro-Manager を自分でインストールしたい場合は、MMI_DownloadMM および MMI_InstallMM の両方で実行する手順を手動で実行する必要があります。

最初のコマンドである MMI_DownloadMM の手順を手動で実行するには、インターネットに接続されたコンピューターを使って、Windows 用の Micro-Manager の適切なバージョンのインストーラーをダウンロードし、そのインストーラーを Igor PC にコピーする必要があります（USB メモリを使うなど）。

インストーラーの URL は現在、下記です。

https://download.micro-manager.org/nightly/2.0/Windows/MMSetup_64bit_2.0.3_20230929.exe

Igor PC に Micro-Manager インストーラーを保存したら、MM_INSTALLER という Windows 環境変数を作成（または変更）し、Igor PC に保存したインストーラーファイルの完全な名前（パスとファイル名）を指定します。絶対パスを使ってください。

相対パスは使わないでください。パス内のディレクトリ名を区切るには、単一のバックスラッシュ文字を使ってください。

Igor が MM_INSTALLER の正しい値を取得するように、環境変数を作成した後、Igor を（再）起動してください。

MMI_InstallMM の操作の手順を手動で実行するには、次の手順が必要です。

1. Micro-Manager インストーラー（MMI_DownloadMM によってダウンロードされたもの、または上記の手順でダウンロードしたもの）を実行し、必要なバージョンの Micro-Manager をインストールしてください。インストーラーを実行するには Windows 管理者権限が必要です。そのため、IT 部門の支援が必要になる場合があります。

Micro-Manager をインストールする時は、Micro-Manager の実行ファイル（例：ImageJ.exe、mmgr_dal_*.dll など）がインストールされているディレクトリの絶対パスを選択し、後でそのパスを指定する必要があります。

Micro-Manager の異なるバージョンは、同一の PC 上に同時に存在させることができます。

したがって、既存の Micro-Manager を削除する必要はありません。

代わりに、Micro-Manager がインストールされるディレクトリに一意の名前を割り当てる方が良いでしょう。

2. Windows の環境変数「MM_BIN」を作成（または編集）し、この変数にこのバージョンの Micro-Manager がインストールされているディレクトリの完全なパスを指定します。
3. Micro-Manager のインストールパスを PATH 環境変数に追加します。これを行うには、Path 変数に%MM_PATH%を追加し、これにより MM_PATH で指定されたディレクトリが自動的に組み込まれるようにします。
4. Igor の MMI サブディレクトリ（通常は「C : ¥Program Files¥WaveMetrics¥Igor Pro 10 Folder¥More Extensions（64-bit）¥Data Acquisition¥MMI」など）に移動し、MMCoreDyn.dll というファイルを MM_BIN で指定されたディレクトリにコピーします。（これには管理者権限が必要になる場合があります。）

Igor の MMI サブディレクトリにある、mmgr_dal_*.dll という形式のファイルも、MM_BIN で指定したディレクトリにコピーし、既存の同名のファイルは上書きしてください。

Igor が MM_BIN 変数の適切な値を使うようにするには、環境変数を作成、編集し、ダイアログを閉じた後に Igor を（再）起動してください。

MMI XOP が正しくインストールされ、Micro-Manager と通信していることを確認するには、次の MMI 関数の戻り値を確認してください。

```
print MMI_GetXOPversion()
```

次のような応答が表示されるはずです。

```
MMI64.xop v.1.1.0.19
```

次のコマンドを実行します。

```
MMI_Reset
```

MMI_GetXOPversion() と MMI_Reset がエラーを発生せずに正常に実行された場合、MMI と Micro-Manager は正しくインストールされ、正常に動作しています。

最後に、Java ベースアプリケーション ImageJ.exe を使う Micro-Manager の標準ユーザーインターフェイスを保持するか削除するかについても検討してください。

詳細は前のセクションを参照してください。

予備的なメモ

注記： Igor Pro は通常、大文字と小文字を区別しませんが、Micro-Manager の多くの部分は、大文字と小文字を区別します。従って、Micro-Manager に渡される文字列（デバイスラベル、デバイス名、プロパティ名、プロパティ値など）の大文字小文字を正しく一致させるように注意する必要があります。MMI への呼び出しが予期せず失敗する場合、Micro-Manager に渡す文字列の大文字小文字の区別を必ず確認してください。

CMMCore クラスの API における関数に対応する MMI 関数およびコマンドの名前は、一般に「MMC_」という接頭辞に続いて、対応する Micro-Manager 関数の名前が続く形式でつけられています。

例えば、loadDevice という CMMCore 関数は、MMC_LoadDevice という Igor コマンドによって公開されています。

MMI のこれらの「コア」関数は、CMMCore によって公開されるパブリック関数を文書化するために Micro-Manager と同じ大まかなカテゴリを使って、次のように示されます。

Micro-Manager の CMMCore クラスが提供する関数の一覧は、以下のリンクから確認できます：

https://micro-manager.org/apidoc/MMCore/latest/class_c_m_m_core.html

MMI は、インターフェイス自体の内部動作を調整するさまざまなコマンドや関数も提供しています。

これらのコマンドと関数の名前は、一般に「MMI_」という接頭辞で始まります。

初期化中に、MMI は、Igor のルートデータフォルダーの下に「MMI」という名前の Igor データフォルダーを作成します。

次に、MMI はその MMI データフォルダーに 3 つのグローバル変数を作成します。

V_MMImresult - MMI はこの数値変数を使って、各コマンドおよび関数の結果を返します。

MMI は、成功を返す時は変数を 0 に設定し、失敗を返す時は非ゼロのエラーコードを設定します。

S_MMImresult - MMI はこの文字列変数を使って、動作中に発生した最新のエラーメッセージをログに記録します。

V_MMImverbose - MMI はこの数値変数を使って、内部エラーの次のように報告方法を決定します：

```
V_MMImverbose = 3          // エラーメッセージを履歴領域に出力して、
                             // エラーコードを Igor に返します。

V_MMImverbose = 2          // エラーメッセージを履歴領域に出力しないが、
                             // エラーコードを Igor に返します。

V_MMImverbose = 1          // エラーメッセージを履歴領域に出力しますが、
                             // エラーコードを Igor に返しません。

V_MMImverbose = 0          // エラーメッセージを履歴領域に出力せず、
                             // エラーコードも Igor に返しません。
```

V_MMImverbose の値は、MMI 内の関数がエラーを報告する方法を明示的に決定します。

一方、MMI のコマンドは、/Q および /Z フラグの状態に応じてエラーを返します。

```
/Q=0 (または指定なし)    // 通常モード：エラーを履歴領域に出力する
/Q=1                      // 静寂モード：エラーを履歴領域に出力しない

/Z=0 (または指定なし)    // 通常モード：エラーを Igor に返す
/Z=1                      // 静寂モード：エラーを Igor に返さない
```

MMI は、V_MMIVerbose の状態や/Q または/Z フラグの有無に関わらず、エラーを V_MMIResult と S_MMIResult に記録します。

MMI を使ってデータを取得する時のパフォーマンスを最適化するためには、画面に表示される「ライブ」画像の数を最小限に抑え、それらの画像が更新される頻度（リフレッシュレート）を低くすると良いです。

このような考慮は、低いフレームレートや小さなサイズの画像（例：ウェブカメラからの映像）のストリーミングには必要ないかもしれませんが、高速または高解像度のカメラからの取得時、または取得中に複雑な画像処理を行う場合、極めて重要となる可能性があります。

画像の取得中に画像の表示方法をコントロールするための複数のオプションが用意されています：

Data Browser を表示する場合、受信する画像ストリームのウェーブのプロットが表示されないように設定することをお勧めします。

Data Browser の Plot チェックボックスをオフにして、ウェーブの波形を表示しないようにするか、メインリストから、受信する画像を含むウェーブ以外の変数またはウェーブを選択してください。

これにより、Data Browser に取得ウェーブが表示されなくなります。

受信側のウェーブをリアルタイムで表示するかどうかを決定できます。

この設定は、Experiment によっては非常に重要ですが、他の Experiment では必要ない場合もあります。

MMI_GetStreamRelayDivisor および MMI_SetStreamRelayDivisor を使って、ストリーミング（シーケンシャル）取得中に、取得した画像のうち実際にウェーブに中継する割合をコントロールすることができます。

リレー除数を 1 より大きい値に設定すると、MMI は $1/n$ 分の画像のみを Igor に送信して分析および表示しますが、これは HDF5 データファイルへの画像の書き込みには影響しません。

これにより、リレー除数を使うことで、MMI のイメージ表示の豊かさ（および負荷）を調整しながら、後々の分析のためにイメージをディスクに保存する方法を変更することなく、表示を最適化できます。

Igor ウェーブでの画像の保存と表示に関する注意

デジタル画像の分野では、一般的に画像のピクセルの行は、画像の上部から下部に向かって数えられ、列は画像の左端から右端に向かって数えられます。

このヘルプでは、これらをそれぞれ「画像行」と「画像列」と呼びます。

Igor は、そのウェーブのデータと列を左から右に、テーブルの上部から下部に向かって、ウェーブ内の行をカウントします。

このノートでは、これらをそれぞれ Igor 行および Igor 列と呼びます。

Micro-Manager が画像を取得して保存する時、その画像のピクセルはコンピューターのメモリ内の最上部の画像行の左端のピクセルから始まる「行優先順序」で配列されます。

メモリ内の連続した位置には、画像行の次のピクセルが格納されており、画像行に沿って左から右へ進み、右端のピクセルに達するまで続きます。

メモリの次の位置には、2 番目の画像行の最も左側のピクセルが格納されており、その後、左から右への進み方が続きます。

対照的に、Igor は 2 次元ウェーブをメモリに格納する場合、Igor 列の最初の値、Igor 行の最初の値から始まる「列優先順」を使います。

メモリ内の連続した位置には、最初の Igor 列の次の値が格納され、Igor 列に沿って上から下へ、最下（つまり最後の Igor 行）に到達するまで進みます。

メモリの次の位置には、2 番目の Igor 列の最も上の値が格納され、その後、上から下へ順に進みます。

この「行順」と「列順」の違いによる結果の一つは、Micro-Manager で取得され、Igor が 2 次元ウェーブとしてメモリに直接保存した画像は、Igor テーブルで表示すると回転して見えることです。

つまり、各画像の行は Igor 列として解釈され、各画像の列は Igor 行として解釈されます。

（この効果は、デジタル画像を Igor ウェーブとして扱う場合によく見られます。Igor のヘルプの「Image Plots」の「Image X and Y Coordinates」を参照してください。）

Igor がウェーブをグラフ化する場合、連続する Igor 行から取得したポイントは、通常、X 軸に沿って水平方向に配列されます。

ウェーブに含まれる値は、通常、Y 軸に沿って垂直方向に表示され、値が大きいほど表示の上部に表示されます。これらの規則は、 $y=f(x)$ という馴染みのある形のグラフを生成します。

一方、デジタル画像が画面に表示（グラフ化）される場合、連続する画像行は通常、垂直に配列され、最初の画像行（行 0）が画面の上部に、最後の画像行が画面の下部に配置されます。

これは垂直軸の反転に相当します。

画像行の座標が低い値ほど表示の上部に近く、座標が高い値ほど表示の下部に近くなります。

従って、Micro-Manager で取得され、Igor が 2 次元ウェーブとしてメモリに直接保存した画像は、デフォルトでは縦軸が反転して表示されます。

（この反転も、Igor ウェーブとしてデジタル画像を扱う場合によく見られます。Igor のヘルプの「Image Plots」の「Image Orientation」を参照してください。）

操作を効率化するため、MMI は Micro-Manager から取得した画像を Igor のウェーブに直接保存し、取得した画像の回転や向き変更は、ユーザーが必要に応じて最適なタイミングで実行できるようにしています。

MMI の開始手順

Micro-Manager は、さまざまなハードウェアへのインターフェイスを提供しています。

これらのインターフェイスは「デバイスアダプター」と呼ばれ、さまざまな人や企業によって作成されており、それぞれ 1 つ以上の異なるタイプのデバイス（1 台以上のカメラ、1 つ以上の光源、フィルターホイールなど）をサポートしています。

デバイスアダプターのリストと使用方法の解説は、<https://micro-manager.org/Device_Support> で確認できます。

Windows では、Micro-Manager のデバイスアダプターは、次のような名前パターンに一致するダイナミック リンク ライブラリ（.dll）ファイルとして実装されています

```
mmgr_dal_<root-name>.dll
```

例えば：

```
mmgr_dal_SutterLambda.dll  
mmgr_dal_OpenCVgrabber.dll
```

メモ：

1. MMI は現在、Micro-Manager ディレクトリツリーのルートにあるファイルを含むデバイスアダプターのみ使用可能です（このディレクトリは MM_BIN 環境変数で指定されます）。
2. マイクロマネージャーのデバイスアダプターは、科学機器でよく必要とされる Windows の「デバイスドライバー」の代わりではありません。
実際には、Micro-Manager のデバイスアダプターは、一般的にそれらのデバイスドライバーに依存しています。
デバイスのハードウェアとデバイスドライバーを正しくインストールすることは、Micro-Manager でデバイスを使うための前提条件です。
Micro-Manager または Igor から新しいデバイスを操作する前に、デバイスの製造元が提供する説明書を確認し、ハードウェアおよび必要なデバイスドライバーがインストールされ、正しく動作していることを確認してください。
3. デバイスアダプターによってサポートされる各デバイスは、デバイスアダプターの作成者によって割り当てられた一意の名前で識別されます。
ユーザーは、各デバイスの各インスタンスに一意のラベルを割り当てる必要があります。

デバイスの1つのインスタンスにラベルを割り当てると、そのインスタンスに対する以降のすべてのコマンドでそのラベルを使います。

ハードウェアを使う時の最初のステップは、1) 適切なデバイスアダプターをロードし、2) 対象のデバイスのインスタンスを作成し、3) そのインスタンスに一意的ラベルを割り当てることです。

MMC_LoadDevice コマンドは、次の手順を実行します。

引数として、そのインスタンスに割り当てるラベル、関連するデバイスアダプターのルート名、および読み込むデバイスの種類名を指定します。

例えば、次の MMC_LoadDevice の呼び出しは、デバイスアダプター「mmgr_dal_DemoCamera.dll」から「DCam」という名前のデモカメラのインスタンスを読み込み、そのインスタンスにラベルを割り当てます：

```
// 使い方 :
// MMC_LoadDevice <device Label>, <root of device adapter>, <name of device-
type>
MMC_LoadDevice "myDemoCam", "DemoCamera", "DCam"
```

読み込まれたデバイスは、後で MMC_UnloadDevice を呼び出すことで解放できます。

デバイスをロードした後、そのデバイスを使う前に、MMC_InitializeDevice を呼び出して、そのハードウェアとインターフェイスを初期化する必要があります。

例えば、

```
// 使い方 :
// MMC_InitializeDevice <device Label>
MMC_InitializeDevice "myDemoCam"
```

最初の例

デバイスを初期化したら、そのデバイスを役割に割り当て、ラベルでデバイスを識別します。

```
// 使い方 :
// MMC_SetCameraDevice <device label>
MMC_SetCameraDevice "myDemoCam"
```

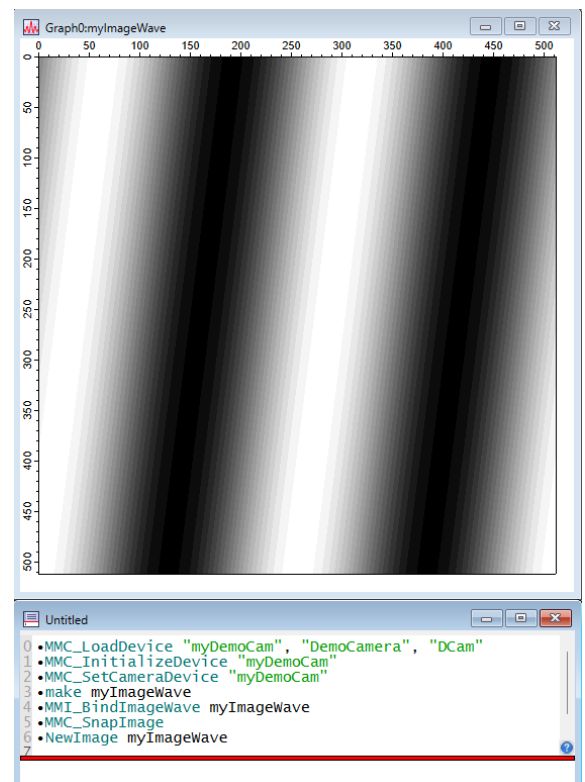
この時点で、Micro-Manager のデモ用カメラは使用可能な状態になっています。

次のステップは、1) 画像を受信するための Igor ウェーブを準備し、2) MMI に、画像の撮影時にそのウェーブを使うように指示することです。

```
// 数値ウェーブを作成
make myImageWave
// MMI に、このウェーブはスナップショットデータを
// 受信するためのものであることを伝える
MMI_BindImageWave myImageWave
// メモ : MMI はカメラからの1つのフレームを
// 保存するためにウェーブの次元を変える
```

ウェーブがバインドされると、MMC_SnapImage を呼び出すことで、写真が撮影され、取得した画像がバインドされた Igor ウェーブに転送されます。

カメラと受信側のウェーブが準備できたら、写真を撮ることができます。



```
// 画像ウェーブがすでにバインドされている場合、  
// このコマンドは画像をウェーブに書き込む
```

```
MMC_SnapImage
```

そして、Igor に画像を表示するように指示します。

```
NewImage myImageWave
```

以上です。

この7つの指示（以下に繰り返します）で、Igor は Micro-Manager に、サポートされているカメラから画像を撮影し、その画像を Igor ウェーブに読み込み、そのウェーブを表示するよう指示することができます。

バインドされた Igor ウェーブが画像のスナップに不要になった場合は、MMI_ReleaseImageWave を呼び出してそれを解放する必要があります。

```
MMC_LoadDevice "myDemoCam", "DemoCamera", "DCam"  
MMC_InitializeDevice "myDemoCam"  
MMC_SetCameraDevice "myDemoCam"  
make myImageWave  
MMI_BindImageWave myImageWave  
MMC_SnapImage  
NewImage myImageWave
```

コンピュータにウェブカメラが接続されている場合、デバイスアダプターと呼ばれる「OpenCVgrabber」を使うことで、そのウェブカメラから画像を取得できる可能性があります。

MMC_LoadDevice への対応する呼び出しは、次のようなものになります。

```
MMC_LoadDevice "myWebCam", "OpenCVgrabber", "OpenCVgrabber"
```

残りの6ステップは上記の例に従い、「myDemoCam」を「myWebCam」に置き換えて実行します。

画像のストリーミング

個々の画像の撮影に加え、Micro-Manager と MMI は「画像のストリーミング」機能（画像の連続的な取得、表示、および（オプションで）保存）をサポートしています。

カメラから画像をストリーミングするには、カメラデバイスをロードし、初期化し、カメラとしての役割に設定する必要があります。

これは、画像の撮影（上記）の場合と同様です。

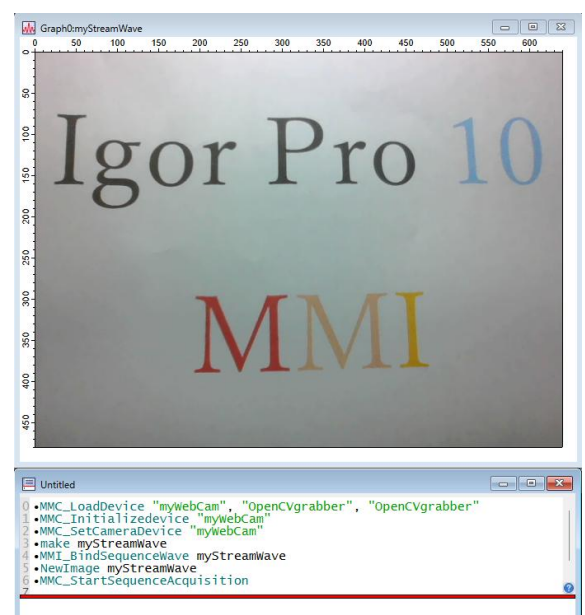
```
MMC_LoadDevice "myWebCam",  
"OpenCVgrabber", "OpenCVgrabber"  
MMC_InitializeDevice "myWebCam"  
MMC_SetCameraDevice "myWebCam"
```

上記と同様に、受信する画像ストリームを受信するための Igor ウェーブを作成し、MMI_BindSequenceWave を呼び出して、このウェーブを画像ストリームに使うことを MMI に通知します。（シングルスナップショットとイメージストリーミングの両方に同じウェーブをバインドすることができます。）

```
make myStreamWave  
MMI_BindSequenceWave myStreamWave
```

Igor 画像ウィンドウにウェーブを表示します。

```
NewImage myStreamWave
```



MMC_StartSequenceAcquisition を呼び出して、ウェーブへの画像のストリーミングを開始します。

```
MMC_StartSequenceAcquisition
```

処理が完了したら、MMC_StopSequenceAcquisition を呼び出してシーケンス（ストリーム）を停止します。

```
MMC_StopSequenceAcquisition
```

従って、8つの Igor ステートメントで、Igor ウェーブへの連続的な画像ストリーミングの初期化、開始、および停止を行うことができます。

ストリーミングにバインドされた Igor ウェーブが不要になった場合は、MMI_ReleaseSequenceWave を呼び出して、そのウェーブを解放する必要があります。

画像とストリームの保存

個別に撮影した画像は、コマンドラインから、または Igor の「データ（Data）」メニューにある「ウェーブを保存（Save Waves）...」サブメニューを開いて、Igor から直接保存することができます。

MMI は、取得中に受信する画像のストリームを直接 HDF5 ファイルに保存するための追加のメカニズムを提供します。

取得終了後にこれらの HDF5 ファイルが正しく閉じられると、Igor で開いて取得後の分析のために再び読み込むことができます。

HDF5 ファイルに関する情報は、<<https://www.hdfgroup.org/solutions/hdf5/>> を参照してください。

画像ストリームを HDF5 ファイルに保存するには、まず「画像のストリーミング」で説明しているように、カメラと MMI を設定して画像ストリームを取得できるようにします。

MMI_OpenH5stream を呼び出し、MMI に HDF5 ファイルを管理するための「HDF5 ストリームハンドラー」を準備させ、画像が到着次第ファイルにルーティングするように指示します：

```
MMI_OpenH5stream // ハンドラーを作成
```

MMI_OpenH5file を呼び出し、ストリームハンドラーに画像を受信するための HDF5 ファイルを作成するように指示します：

```
variable V_myH5fileID // ファイルの ID を保存
MMI_OpenH5file V_myH5fileID, "C:\\Users\\Igor\\Desktop\\webcam_stream.h5"
```

Windows のパス表記を使う場合、パス内の名前を区切るには必ず「二重の」バックスラッシュ文字を使ってください。

Igor は MacOS と長い付き合いがあるため、コロン「:」でパス内の名前を区切る Macintosh HFS パス表記も使うことができます。

詳細については、MMI_OpenH5file を参照してください。

MMI_OpenH5group を呼び出して、ストリームハンドラーにファイル内に HDF5 グループを作成し、画像を受け取るように指示します：

```
variable V_myH5groupID // グループの ID を保存
MMI_OpenH5group V_myH5groupID, "firstGroup" // グループ名はオプション
```

ストリームハンドラーを起動し、ファイルとグループを開いた後、MMC_StartSequenceAcquisition と MMC_StopSequenceAcquisition を呼び出すことで、ストリームの取得を開始および停止できます。

MMI は、可能な限り、受信した各画像をアクティブファイルのアクティブグループに、それぞれ独立したデータセットとして追加します。

グループへの取得が完了次第、MMI_CloseH5group を呼び出してグループを閉じ、誤って破損するリスクを軽減してください。

```
MMI_CloseH5group V_myH5groupID
```

この時点で以前のグループは閉じられますが、新しいグループを作成し、そのグループに一意の名前を付けることで、ファイルへの書き込みを続けることができます。

```
MMI_OpenH5group V_myH5groupID, "secondGroup"
```

この 2 番目のグループも、その取得が完了した時点で閉じる必要があります：

```
MMI_CloseH5group V_myH5groupID
```

ファイルへのすべての読み込みが完了したら、同様にファイルを閉じます：

```
MMI_CloseH5file V_myH5fileID
```

この時点で、指定された HDF5 ファイル内の各グループには、ストリーム化された画像の 1 つを含むデータセットの配列が格納されています。

各グループの HDF5 属性には、グループが開かれた時点でのカメラのプロパティ設定が含まれており、各データセットの属性には、各画像に割り当てられた Micro-Manager のメタデータが含まれています。

Igor のコマンドライン機能、または Igor の Data メニューの Load Waves サブメニューにある HDF5 ブラウザを使って、任意の 1 つまたはすべての画像を Igor ウェーブで開くことができます。

この Igor 関数では、カメラの初期化から、ストリームおよび HDF5 コンポーネントの準備、ストリームの取得、HDF5 コンポーネントの閉じるまでのプロセスが示されています。

操作の順序を明確にするため、このサンプルコードではエラーチェックを行いません。

コードにエラーチェックを含めることをおすすめします。

```
Function MMI_HDF5demo(variable numFrames, string fileName)
variable v_h5fileID, v_h5groupID
    MMI_Reset // この例をクリーンスタート

    // カメラを準備
    MMC_LoadDevice "myWebCam", "OpenCVgrabber", "OpenCVgrabber"
    MMC_Initializeddevice "myWebCam"
    MMC_SetCameraDevice "myWebCam"

    // ストリーミングウェーブを準備
    make /O m_img
    MMI_BindSequenceWave m_img
    newimage m_img // このステップはオプション、HDF5 に書き込むには必要ない

    // HDF5 出力を準備
    mmiopenh5stream
    mmiopenh5file /o v_h5fileID, fileName
    mmiOpenH5group v_h5groupID, "group1"

    // 取得の開始
    mmc_startsequenceAcquisition /Q /N=(numFrames)

    // 取得終了を待機
    do
        sleep /t 1
    while (MMC_isSequenceRunning())

    // HDF5 処理をクローズ
    mmi_CloseH5group v_H5groupID
    mmi_CloseH5file v_H5fileID
    mmi_CloseH5stream
```

次のステップ

Micro-Manager 内のすべてのデバイスには、そのデバイスの動作を説明し詳細化する「プロパティ」のリストが関連付けられています。

一部のプロパティの値を調整（または設定）できます。

このようなプロパティは「書き込み可能」または「読み書き可能」と呼ばれます。

他のプロパティは変更できません。それらの値は読み取り専用（取得のみ可能）です。

デバイスのプロパティを読み書きするには、まず MMC_LoadDevice を使ってデバイスをロードし、MMC_InitializeDevice を使って初期化する必要があります。

これらの手順が完了したら、MMI_GetDevicePropertyList を呼び出して、デバイスのプロパティ（名前、説明、値、制限、データ形式）のリストを含む 2 次元 Igor テキストウェーブを取得できます。

MMC_GetPropertyNumeric および MMC_GetPropertyString 関数を使うと、任意の 1 つのプロパティの現在の値を取得できます。

「読み書き可能」なプロパティの値は、MMC_SetProperty コマンドを使って設定できます。

インターフェイスのサマリ

管理インターフェイス

MMI_DownloadMM	MMI の動作バージョンに対応した Micro-Manager のバージョン用の Windows インストーラーをダウンロードし、保存します。
MMI_InstallMM	MMI_DownloadMM によってダウンロードされた Windows インストーラーを実行し、コンピュータに Micro-Manager をインストールし、Micro-Manager が MMI と動作するように準備します。
MMI_GetXOPversion	MMI XOP のバージョン番号を含む Igor 文字列を返します。
MMI_GetDevicePropertyList	Igor テキストウェーブを返します。
MMI_Reset	Igor と Micro-Manager の間のインターフェイスをリセットし、バインドされているすべてのウェーブを解放し、開いているすべてのファイルを閉じます。 また、Micro-Manager コア、ストリームマネージャー、およびデフォルトのストリームシンクを解放し、再作成します。
MMI_GetStreamRelayDivisor	Igor ストリームマネージャーが、アクティブなストリームから受信したフレームを、受信先の Igor ウェーブに中継する頻度 (1/n) を返します。
MMI_SetStreamRelayDivisor	Igor ストリームマネージャーが、アクティブなストリームから受信したフレームを、受信先の Igor ウェーブに中継する頻度 (1/n) を設定します。
MMI_BindImageWave	MMC_SnapImage の呼び出しによって取得された画像を受信するために、指定された Igor ウェーブを保持するように XOP に指示します。
MMI_ReleaseImageWave	MMC_SnapImage の呼び出しによって取得された画像を受信するために保持されている Igor ウェーブを解放するように XOP に指示します。

MMI_BindSequenceWave	MMC_StartSequenceAcquisition を使って、順番に取得された画像を受信するために、指定された Igor ウェーブを保持するように XOP に指示します。
MMI_ReleaseSequenceWave	ストリーミング操作によって取得された画像を受信するために保持されている Igor ウェーブを解放するように XOP に指示します。
MMI_BindImageProcessor	XOP に、各画像の取得後に処理するために呼び出すべきユーザーが提供した関数を指定します。
MMI_ReleaseImageProcessor	XOP に、各画像の取得後にユーザーが提供した関数を呼び出さないように指示します。
MMI_BindMetaDataWave	XOP に、順番に取得される画像からメタデータを受信するために、指定された Igor テキストウェーブを保持するように指示します。
MMI_ReleaseMetaDataWave	XOP に、順番に取得された画像からメタデータを受信するために保持されている Igor テキストウェーブを解放するように指示します。
MMI_WaitForSequenceCompletion	XOP に、Micro-Manager が連続取得を実行中でないまで待機するように指示します。
MMI_OpenH5stream	HDF5 ストリームのインスタンスが存在しない場合、そのインスタンスを作成し初期化します。
MMI_CloseH5stream	現在の HDF5 ストリームを閉じ、存在する場合に削除します。 その際、現在開かれているすべてのファイルまたはグループを最初に閉じます。
MMI_IsH5streamOpen	HDF5 ストリームがオープンされているかどうかを報告します。
MMI_OpenH5file	HDF5 ストリームを操作する関数で、指定された名前で HDF5 ファイルを開き、一意のファイル ID を返します。
MMI_CloseH5file	指定されたファイル ID で識別される HDF5 ファイルを閉じるように HDF5 ストリームに指示します。
MMI_SelectH5file	HDF5 ストリーマーに、指定された ID で識別される HDF5 ファイルを「アクティブ」に設定するように指示します。
MMI_OpenH5group	HDF5 ストリーマーに、以前に開いた HDF5 ファイル内の HDF5 グループを開くように指示します。 一意のグループ ID を返します。
MMI_CloseH5group	指定されたグループ ID で識別される HDF5 グループを閉じるように HDF5 ストリーマーに指示します。
MMI_SelectH5group	HDF5 ストリーマーに、指定されたグループ ID で識別される HDF5 グループを「アクティブ」に設定するように指示します。
MMI_GetDevicePropertyList	指定されたラベルを持つデバイスのプロパティ一覧を取得します。 リストには、プロパティの名前、値、型、変更可能かどうか、および（該当する場合）最小値と最大値が含まれます。
MMI_GetEstMaxFrameCount	MMI がメモリに直接取得して追加できる最大フレーム数を返します。
MMI_GetCamera	Igor の以前バージョンの GetCamera コマンドを置き換えます。
MMI_NewCamera	Igor の以前バージョンの NewCamera コマンドを置き換えます。
MMI_ModifyCamera	Igor の以前バージョンの ModifyCamera コマンドを置き換えます。

Micro-Manager コアインターフェイス

初期化とセットアップ

MMC_LoadDevice	サポートされているハードウェア（カメラ、シャッター、フィルターホイール、光源など）用の新しいソフトウェアデバイスインターフェイスを作成し、識別ラベルを割り当てます。
MMC_UnloadDevice	MMC_LoadDevice より前の呼び出しによって作成されたデバイスインターフェイスを解放します。
MMC_UnloadAllDevices	これまでに作成されたすべてのデバイスインターフェイスを解放します。
MMC_InitializeAllDevices	これまでに作成されたすべてのデバイスインターフェイスと、それに対応するハードウェアを初期化します。
MMC_InitializeDevice	指定されたデバイスラベルに関連付けられたデバイスインターフェイスおよび対応するハードウェアを初期化します。
MMC_Reset	Micro-Manager のコアをリセットし、これまでに作成されたすべてのデバイスインターフェイスを解放します。
MMC_UpdateCoreProperties	Micro-Manager のコアプロパティ（割り当てられたデバイスのセット）を、現在読み込まれているデバイスを使って更新します。
MMC_GetCoreErrorText	指定された数値のエラーコードに対応する Micro-Manager のエラーテキストを返します。
MMC_GetVersionInfo	Micro-Manager のコアのバージョン番号を返します。
MMC_GetAPIVersionInfo	Micro-Manager の Module-API と Device-API のバージョン番号を返します。
MMC_SaveSysState	システムの現在の状態（つまり、書き込み可能なプロパティの状態）を指定されたファイルに保存します。
MMC_LoadSysState	指定されたファイルからシステムの現在の状態（つまり、書き込み可能なプロパティの状態）を読み込みます。
MMC_SaveSysConfig	システムの現在の設定（読み込まれたデバイス、そのラベルなど）を指定されたファイルに保存します。
MMC_LoadSysConfig	指定されたファイルから、システムの現在の設定（読み込まれているデバイス、そのラベルなど）をインストールします。

ログ記録とログ管理

MMC_SetPrimaryLogFile	指定されたファイルを Micro-Manager の主ログファイルとして設定します。
MMC_GetPrimaryLogFile	Micro-Manager の主ログファイルの名前を含む文字列を返します。
MMC_LogMessage	Micro-Manager の主ログファイルにメッセージを追加します。
MMC_EnableDebugLog	Micro-Manager のデバッグメッセージのログ記録を有効または無効にします。

MMC_DebugLogEnabled	デバッグメッセージのログ記録が有効になっているかどうかを示す数値を返します。
MMC_EnableStdErrLog	Micro-Manager のログメッセージを「標準エラー」(stderr) ファイルに送信する機能を有効または無効にします。これは主に Micro-Manager のコード開発者向けです。
MMC_StdErrLogEnabled	ログメッセージが「標準エラー」(stderr) ファイルに送信されるかどうかを示す数値を返します。これは主に Micro-Manager のコード開発者向けです。
MMC_StartSecondaryLogFile	Micro-Manager がログ出力を追加のファイルに記録し始めます。
MMC_StopSecondaryLogFile	Micro-Manager がログ出力を追加のファイルにキャプチャするのを停止します。

デバイスのリスト

MMC_GetAvailableDevices	指定されたモジュール (デバイスアダプター) から利用可能なデバイスの名前の一覧を取得し、デバイスごとにそのデバイスタイプと説明を付加します。
MMC_GetLoadedDevices	現在読み込まれているデバイスの名前の一覧を取得します。
MMC_GetDeviceAdapterNames	検出可能なすべてのデバイスアダプターのルート名の一覧を取得します。

汎用のデバイスコントロール

MMC_GetDeviceType	指定されたラベルを持つデバイスのデバイスタイプを返します。
MMC_GetDeviceName	指定されたラベルを持つデバイスのデバイス名を返します。
MMC_GetDeviceLibrary	指定されたラベルを持つデバイスのデバイスアダプターのルート名を返します。
MMC_GetPropertyString	指定されたラベルを持つデバイスに対して、そのデバイスに定義された名前付きプロパティの値を文字列として返します。
MMC_GetPropertyNumeric	指定されたラベルを持つデバイスに対して、そのデバイスに定義された名前付きプロパティの値を数値として返します。
MMC_SetProperty	指定されたラベルのデバイスに対して、指定されたプロパティの値を設定します。
MMC_GetAllowedPropertyValues	指定されたラベルを持つデバイスの名前付き文字列プロパティに対して、許可されたテキスト値のリストを取得します。
MMC_IsPropertyReadOnly	指定されたデバイスの指定されたプロパティが読み取り専用プロパティであるかどうかを返します。
MMC_IsPropertyPreInit	指定されたデバイスの指定されたプロパティが初期化前に定義されていなければならないかどうかを返します。

特定の役割に対する現在のデバイスの管理

MMC_GetCameraDevice	アクティブなカメラの Micro-Manager のラベルを返します。
MMC_SetCameraDevice	現在アクティブなカメラを設定します。
MMC_GetShutterDevice	アクティブなシャッターに対応する Micro-Manager ラベルを返します。

MMC_SetShutterDevice	現在アクティブなシャッターを設定します。
MMC_GetFocusDevice	アクティブなフォーカスデバイスに対応する Micro-Manager ラベルを返します。
MMC_SetFocusDevice	現在アクティブなフォーカスデバイスを設定します。
MMC_GetXYStageDevice	アクティブな XY ステージの Micro-Manager ラベルを返します。
MMC_SetXYStageDevice	アクティブな XY ステージを設定します。
MMC_GetAutoFocusDevice	アクティブなオートフォーカスデバイスの Micro-Manager ラベルを返します。
MMC_SetAutoFocusDevice	現在アクティブなオートフォーカスデバイスを設定します。
MMC_GetSLMDevice	アクティブな空間光モジュレーター（SLM）の Micro-Manager ラベルを返します。
MMC_SetSLMDevice	アクティブな空間光モジュレーター（SLM）を設定します。
MMC_SetGalvoDevice	アクティブなガルバノメータ光路偏向装置の Micro-Manager ラベルを返します。
MMC_GetGalvoDevice	アクティブなガルバノメータ光路偏向装置を設定します。
画像の取得	
MMC_GetROI	指定したカメラの対象領域（ROI）を含む Igor ウェーブを返します。
MMC_SetROI	指定した Igor ウェーブから、指定したカメラの ROI（Region Of Interest）を設定します。
MMC_ClearROI	アクティブなカメラの ROI をフルセンサーに設定します。
MMC_IsMultiROISupported	アクティブなカメラが複数の ROI の使用をサポートしているかどうかを返します。
MMC_IsMultiROIEnabled	アクティブなカメラが現在、複数の ROI を使用しているかどうかを返します。
MMC_GetMultiROI	アクティブなカメラで使用中の ROI のセットを表すウェーブを返します。
MMC_SetMultiROI	アクティブカメラで使用する ROI のセットを定義します。
MMC_GetNumberOfComponents	アクティブなカメラが取得したコンポーネント（色チャンネル）の数を返します。
MMC_GetExposure	現在アクティブなカメラの露出時間をミリ秒単位で返します。
MMC_SetExposure	アクティブなカメラの露光時間をミリ秒単位で設定します。
MMC_InitializeCircularBuffer	アクティブなカメラから送信される画像の寸法に基づいて、Micro-Manager の循環バッファをリセットします。
MMC_ClearCircularBuffer	Micro-Manager の循環バッファの内容をクリアし、存在する可能性のある画像をすべて削除します。
MMC_GetCircularBufferSize	Micro-Manager の循環バッファのサイズを MB 単位で返します。
MMC_SetCircularBufferSize	Micro-Manager の循環バッファのサイズを MB 単位で設定します。

MMC_SetAutoShutter	画像の取得時にアクティブなシャッターの自動開閉を有効または無効にします。
MMC_SetShutterOpen	指定されたシャッターの開閉状態を設定します。
MMC_GetImageWidth	画像バッファの水平方向のサイズ（つまり列の数）をピクセル単位で返します。
MMC_GetImageHeight	画像バッファの垂直方向のサイズ（つまり行数）をピクセル単位で返します。
MMC_GetImageBitDepth	各画像ピクセルの明るさをエンコードするために使用されたビット数を返します。
MMC_SnapImage	現在アクティブなカメラから 1 枚の画像を取得します。
MMC_GetImage	これまでに取得した画像を取り出し、ウェーブに書き込みます。
MMI_GrabImage	1 枚の画像を取得し、MMC_GetImage よりも細かい制御でウェーブに書き込みます。
MMC_PrepareSequenceAcquisition	カメラを連続撮影の開始に備えて準備します。
MMC_StartSequenceAcquisition	連続取得を開始します。
MMC_StopSequenceAcquisition	これまでに開始された連続取得を停止します。
MMC_IsSequenceRunning	Micro-Manager が現在、連続取得を実行中かどうかを報告します。
デバイスコントロールの状態	
MMC_GetState	Micro-Manager が現在、連続取得を実行中かどうかを返します。
MMC_DefineStateLabel	指定されたデバイスの指定された状態に対して、状態ラベル（文字列）を定義します。
MMC_SetState	指定されたデバイスの状態を、指定された数値の状態に設定します。
MMC_SetStateLabel	指定されたデバイスの状態を、指定された状態ラベルに対して事前に定義された数値状態に設定します。
ハブと周辺機器	
MMC_GetParentLabel	親デバイスのラベルを返すことで、Micro-Manager デバイスの「親」を特定します。
MMC_SetParentLabel	親デバイスのラベルを指定することで、Micro-Manager デバイスをその「親」にバインドします。

インターフェイスのメソッド

MMI_DownloadMM /Q /Z /FILE=*filename* /SRC=*pathname* /URL=*sourceURL* /DEST=*destPath*

MMI の動作バージョンに対応した Micro-Manager のバージョン用の Windows インストーラーをダウンロードし、保存します。

フラグ

- /Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
- /Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。
- /R 「レポート」モードは、MMI にデフォルトの URL（通常は Micro-Manager インストーラーをダウンロードする URL）を単に返すように指示し、ダウンロードは行いません。
MMI は、Igor の履歴と MMI データフォルダーのグローバル変数 S_MMImresult に文字列を書き込むことで URL を返します。
- /FILE=fileName MMI 内に保存されているデフォルトの名前を上書きし、ダウンロードするインストールファイルの名前を指定します。
- /SRC=pathname Windows 状の Micro-Manager インストーラーのローカルコピーの絶対パスを指定します。
/SRC オプションは、/URL および /FILE オプションよりも優先されます。
- /URL=sourceURL MMI 内に保存されているデフォルトの名前を上書きし、ファイルをダウンロードするインターネット URL の名前を指定します。
- /DEST=destPath ダウンロードしたファイルを保存するディレクトリのパスを指定します。これにより、MMI が通常選択するデフォルトの保存先ディレクトリ（Windows 環境変数で設定された DESKTOP または TEMP ディレクトリ）が上書きされます。

詳細

フラグが指定されていない場合、XOP は Igor の URLRequest コマンドを呼び出してダウンロードを実行し、受信したファイルを保存します。

インターネットから隔離されたコンピューターに MMI をインストールするユーザーは、インターネットに接続された別のコンピューターを使って Micro-Manager インストーラーをダウンロードし、その後、そのインストーラーをインターネットから隔離されたコンピューターに移動させることができます（例：USB メモリを使う）。

このような場合、/SRC オプションを使って、USB メモリ内のインストーラーのパスを指定します。

参照

MMI_InstallMM

MMI_InstallMM /Q /Z /RJ /DIR=destFolder

MMI_DownloadMM によってダウンロードされた Windows インストーラーを実行し、Micro-Manager をコンピューターにインストールし、MMI で Micro-Manager を使用できるように準備します。

フラグ

- /Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

/RJ 「Java を削除」オプションは、MMI に Micro-Manager を通常どおりインストールした後、インストール完了後に Micro-Manager の Java コンポーネントを削除するよう指示します。
(これは主に、Java コードの実行に伴うセキュリティ上の影響を懸念する研究機関や開発チームを対象としています)
ただし、/RJ オプションを選択すると、この Micro-Manager のインストールをスタンダードプログラムとして実行することはできなくなります。

/DIR=destFolder
Micro-Manager の新規インストールのルートとなるディレクトリのパスを指定します。
指定した場合、PROGRAMFILES 環境変数と Windows インストーラーの名前によって決定されるデフォルトの場所が上書きされます。

詳細

XOP は、Igor の URLRequest コマンドを呼び出してダウンロードを実行し、受信したファイルを保存します。

参照

MMI_DownloadMM

MMI_GetXOPversion ()

MMI64.xop の現在実行中のコピーのバージョン番号を含む Igor 文字列を返します。

パラメーター

(なし)

例

次のコマンドを入力すると：

```
print MMI_GetXOPversion()
```

次の形式返されます：

```
MMI64.xop v.1.1.0.8
```

XOP のバージョンは、Micro-Manager コアのバージョン番号や Micro-Manager API のバージョン番号からは独立しています。

参照

MMC_GetVersionInfo、MMC_GetAPIVersionInfo

MMI_Reset

MMI インターフェイスを完全にリセットします。

MMI_Reset は、実行中のシーケンスをすべて停止し、バインドされているウェーブ（画像ウェーブ、ストリームウェーブ、メタデータウェーブ）をすべて解放し、バインドされている処理関数をすべて解放し、ストリームマネージャーを終了し、Micro-Manager コアをリセットして解放し、すべてのデバイスを解放します。

その後、Micro-Manager コアとストリームマネージャーの新しいインスタンスを作成します。

参照

MMC_Reset、MMI_ReleaseImageWave、MMI_ReleaseSequenceWave、
MMI_ReleaseMetaDataWave、MMI_ReleaseImageProcessor

MMI_GetDevicePropertyList /Q /Z deviceLabel [, recipientWave, prefix]

指定されたラベルを持つデバイスのプロパティ一覧を取得します。

一覧には、プロパティの名前、値、タイプ、変更可能かどうか、（該当する場合）最小値と最大値が含まれます。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsrlt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

deviceLabel 選択されたデバイスの大文字小文字を区別する Micro-Manager ラベルを含む文字列です。

recipientWave 選択したデバイスのプロパティのリストを受信するテキストウェーブのオプション名です。
このパラメーターを省略した場合、MMI は現在のデータフォルダーに「M_DeviceProperties」というベース名を使ってテキストウェーブを作成します。

prefix ウェーブの各要素にプレフィックスとして追加するテキストを含むオプションの文字列です。
これにより、ウェーブの各要素に注釈エスケープコードを簡単に挿入でき、リストボックスなどの Igor のコントロールで表示する時にテキストを簡単にフォーマットすることができます。

参照

MMC_SetProperty、MMC_GetPropertyString、MMC_GetPropertyNumeric

MMI_GetStreamRelayDivisor ()

入力画像ストリーム内の画像が指定された Igor ウェーブに渡される頻度を反映した数値返します。

パラメーター

(なし)

詳細

返される整数値は、 $1/n$ の形の分数の分母として解釈され、 n フレームのうち 1 フレームだけが Igor ウェーブに渡されることを示します。

従って、戻り値が 2 であれば、2 フレームごとにフレームが Igor に渡されることを意味します。

戻り値が 10 の場合は、受信した 10 フレームのうち 1 フレームだけが Igor に渡されます。

参照

MMI_SetStreamRelayDivisor

MMI_SetStreamRelayDivisor /Q /Z newValue

受信画像ストリーム内の画像が、選択した Igor ウェーブに渡される頻度を設定します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

newValue 1 から $(2^{32})-1$ までの数値です。
これは $1/n$ の形の分数の分母として解釈され、Igor ウェーブに「 n フレームのうち 1 フレームのみ」が渡されることを示します。

詳細

ストリームリレーディバイダー (Stream Relay Divisor) を 2 に設定すると、2 枚ごとに画像が受信側のウェーブに渡されます。

分母に 10 を設定すると、MMI は 10 番目のウェーブごとにのみ受信ウェーブに渡すよう指示します。受信画像の一部だけを Igor の受信ウェーブに中継することで、Igor が画像の更新に費やす時間を短縮でき、取得や受信ストリームのファイルへの保存に、より多くの時間を割くことができます。

起動時またはリセット時にインストールされる分母のデフォルト値は 1 です。

許容値は 1 から $(2^{32})-1$ までの正の整数です。

ストリーム リレーディバイダーは HDF5 ストリーマーに直接影響しないことに注意してください。順に取得 (ストリーム) された画像は、リレーディバイダーの値に関わらず、すべて HDF5 ストリーマーにリレーされます。

参照

MMI_GetStreamRelayDivisor、MMI_OpenH5stream

MMI_OpenH5stream /Q /Z

HDF5 ストリームがまだ開かれていない場合、HDF5 ストリームを作成し初期化します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsrlt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

(なし)

詳細

最高のパフォーマンスを維持するため、XOP は HDF5 ストリームを管理し、連続取得された（ストリーミングされた）画像を HDF5 ファイルに書き込むための HDF5 ストリーマーのインスタンスを自動的に作成しません。

従って、ユーザーは HDF5 ファイルにストリーミングされた画像を保存する前に、HDF5 ストリーマーのインスタンスを作成する必要があります。

ストリームリレーディバイダーは、取得中にストリーミングされた画像が Igor にリレーされる頻度のみを変更します。

ストリームリレーディバイダーは、ストリーミングされた画像が HDF5 ファイルに書き込まれる方法に影響を与えません。

連続取得（ストリーミング）された各画像は、リレーディバイダーの値に関わらず、HDF5 ストリームラーに送信されます。

参照

MMI_CloseH5stream、MMI_IsH5streamOpen、MMI_GetStreamRelayDivisor、
MMI_SetStreamRelayDivisor

MMI_CloseH5stream /Q /Z

存在する場合、HDF5 ストリーマーを閉じ、削除します。最初に、開いているすべての HDF5 ファイルとグループを閉じます。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsrlt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

(なし)

詳細

MMI_CloseH5Stream を呼び出すと、開いているすべての HDF5 グループとファイルが自動的に閉じられます。

従って、各グループとファイルを個別に閉じる代わりに、MMI_CloseH5Stream を呼び出す方が簡単かもしれません。

ただし、特定のグループまたはファイルを閉じたいが、すべてを閉じる必要がない場合は、そのグループとファイルに対してのみ、MMI_CloseH5group と MMI_CloseH5file を個別に呼び出す必要があります。

参照

MMI_OpenH5stream、MMI_IsH5streamOpen、MMI_CloseH5file、MMI_CloseH5group

MMI_IsH5streamOpen ()

HDF5 ストリーマーが現在開かれているかどうかを数値で返します。

パラメーター

(なし)

戻り値

HDF5 ストリーマーが開いている場合、1 を返します。

HDF5 ストリーマーが閉じている場合、0 を返します。

参照

MMI_OpenH5stream、MMI_CloseH5stream

MMI_OpenH5file /Q /Z /O fileID, fileName

指定されたファイル名で新しい HDF5 ファイルを作成し、そのファイルを現在アクティブなファイルとして選択し、一意のファイル ID を返します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

/O 「上書き」モードは、MMI に同じ名前を持つ既存のファイルを置き換えるように指示します。
指定されたファイルが存在し、/O フラグが指定されていない場合、既存のファイルが再オープンされ、以降の MMI_OpenH5group への呼び出しは、既存のファイルに新しいグループを付加します。

パラメーター

fileID 新規に作成された HDF5 ファイルに割り当てられる ID 番号を受け取る、実際の（複素数ではない）数値変数です。
ファイル ID 番号は、1 以上の一意の整数です。

filename HDF5 ファイルを作成するためのパスとファイル名を指定する文字列です。
シンプルなファイル名または相対パスは、ファイル MMI64.xop の場所（通常は c:¥Program Files¥Wavemetrics¥Igor 10 Pro Folder¥Igor Extensions (64-bit)）に対して解釈されます。
目的のファイルを指定する際は、フルパスを使うことを推奨します。

Windows のパス区切り文字を使う場合、バックスラッシュ文字を「二重」に指定する必要があります。例：

```
MMI_OpenH5file v_FileID, "c:\\temp\\myHDF5file.h5"
```

MMI は、Macintosh の HFS パス区切り文字の使用もサポートしています（ただし、MMI は Windows 環境下でのみ動作します）。例：

```
MMI_OpenH5file v_FileID, "c:temp:myHDF5file.h5"
```

詳細については、パス区切り記号を参照してください。

詳細

MMI_OpenH5file は、入力される画像が現在アクティブなカメラの現在の設定を使って取得されるものと仮定します。

このコマンドにより、カメラから取得する画像の形式（例：画像の幅と高さ、ピクセルあたりのビット数、色平面の数など）に内部的に対応する準備が行われます。

MMI では、異なるカメラ（または異なる設定）から取得した画像を、別々の HDF5 ファイルに書き込むか、または指定された HDF5 ファイルを閉じてから、別のアクティブなカメラで再度開く必要があります。

HDF5 ファイルを特定のカメラ用に設定するには、MMI_OpenH5file を呼び出す前に、MMC_SetCameraDevice を呼び出してアクティブなカメラを設定してください。

ファイルを開いたときにアクティブだったものとは異なるカメラや設定からの HDF5 ファイルに画像をストリーミングする場合、不正な結果が発生する可能性があります。

アクティブなカメラとアクティブな HDF5 ファイルを同期させるには、MMC_SetCameraDevice と MMI_SelectH5file を組み合わせて使うようにしてください。

複数の HDF5 ファイルとグループ（下記参照）を同時に開くことは可能ですが、一度にアクティブにできるのは 1 つのファイルと 1 つのグループのみです。

Windows は特定のフォルダー（c:\Program Files や c:\Windows など）の内容を保護し、通常、アプリがそれらのフォルダー（またはそのサブフォルダー）内にファイルを作成または変更することを禁止しています。

従って、Windows で保護されているフォルダー内に HDF5 ファイルを作成しようとする、MMI_OpenH5file は失敗する可能性が高いです。

このような問題が発生しないように、Igor の SpecialDirPath 関数を使うことを検討してください。

この関数は、複数の書き込み可能なフォルダーへのパスを返すことができます。

参照

MMC_SetCameraDevice、MMI_CloseH5file、MMI_SelectH5file、MMI_OpenH5group、パス区切り記号、SpecialDirPath

MMI_CloseH5file /Q /Z fileID

MMI_OpenH5file の以前の呼び出しで開いた HDF5 ファイルを閉じます。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

fileID MMI_OpenH5file 関数によって HDF5 ファイルに割り当てられた ID 番号です。
ファイル ID 番号は、1 以上である整数でなければなりません。

詳細

MMI_CloseH5file は、そのファイル内で現在開かれているすべての HDF5 グループを自動的に閉じます。

参照

MMI_OpenH5file、MMI_SelectH5file

MMI_SelectH5file /Q /Z fileID

MMI_OpenH5file 関数によって以前に開かれた HDF5 ファイルを、現在アクティブな HDF5 ファイルとして選択します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

fileID 選択するファイルの ID 番号です。
fileID は、MMI_OpenH5file 関数による呼び出しで HDF5 ファイルに事前に割り当てられていなければなりません。

詳細

複数の HDF5 ファイルとグループ（下記参照）を同時に開くことは可能ですが、一度にアクティブにできるのは1つのファイルと1つのグループのみです。

参照

MMI_OpenH5file、MMI_CloseH5file、MMI_OpenH5group

MMI_OpenH5group /Q /Z /F=H5fileID groupID [, groupName]

開いている HDF5 ファイル内に HDF5 グループを作成し、そのグループをファイルのアクティブグループとして選択し、指定されたファイル内の新しいグループに割り当てられた ID 番号を返します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

/F=H5fileID	グループを作成する HDF5 ファイルの ID 番号です。 これは、MMI_OpenH5file コマンドによって返されるファイル ID です。 /F フラグを使っても、指定された HDF5 ファイルをアクティブなファイルとして選択しません。 指定されたファイル内に HDF5 グループを作成するだけです。 アクティブな HDF5 ファイルを選択するには、必ず MMI_SelectH5file を呼び出してください。
groupID	新規に作成される HDF5 グループに割り当てられる ID 番号を受け取る、実際の（複素数ではない）数値変数です。 グループ ID 番号は、1 以上である整数です。
groupName	オプションの文字列で、新規に作成されるグループに割り当てる一意の名前を指定します。 groupName が指定されていない場合、コマンドは新しいグループを作成し、以下の通り一意の名前を割り当てます。 指定されたファイル内では groupName は一意でなければなりませんが、異なるファイルでは同じグループ名を使用できます。

詳細

複数の HDF5 グループとファイルが同時に開かれている場合でも、一度にアクティブにできるのは 1 つのファイルと 1 つのグループのみです。

HDF5 グループが閉じられると、もう一度開くことはできません。

コマンドによって作成される名前は、コマンドが実行される時に生成される Windows UUID を基に作成されます。

作成された名前は「Group_xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx」の形式になります。

MMI_OpenH5group 関数によって返される groupID は、その HDF5 ファイル内では一意ですが、すべての開いている HDF5 ファイル間で一意であることは保証されないことに注意してください。

つまり、複数の開かれた HDF5 ファイルには、それぞれ同じグループ ID を持つグループが含まれている可能性があります。

複数の開いている HDF5 ファイルの中から特定のグループをユニークに指定するには、必ずファイル ID とグループ ID の両方を使ってください。

参照

MMI_OpenH5file、MMI_SelectH5file、MMI_CloseH5group、MMI_SelectH5group、
[<https://en.wikipedia.org/wiki/Universally_unique_identifier>](https://en.wikipedia.org/wiki/Universally_unique_identifier)

MMI_CloseH5group /Q /Z /F=H5fileID groupID

MMI_OpenH5group コマンドによって以前に作成された HDF5 グループを閉じます。

フラグ

/Q	「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
/Z	「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。 /Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。 エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。
/F=H5fileID	閉じて削除するグループを含む HDF5 ファイルの ID 番号です。 これは、MMI_OpenH5file コマンドによって返されるファイル ID です。

パラメーター

groupID HDF5 グループに以前割り当てられた ID 番号です。
これは MMI_OpenH5group によって返されるグループ ID です。

詳細

MMI は現在、一度閉じられた HDF5 グループを書き込み用に再オープンできません。
HDF5 グループは、後で読み取り用に再オープンされる可能性があります。

参照

MMI_OpenH5group、MMI_SelectH5group、Igor Pro の HDF5

MMI_SelectH5group /Q /Z /F=H5fileID groupID

HDF5 グループ（およびオプションで HDF5 ファイル）を選択し、現在アクティブなグループ（およびファイル）として設定します。

ファイルとグループは、事前に MMI_OpenH5file および MMI_OpenH5group の呼び出しによって開かれていなければなりません。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

/F=H5fileID 選択するグループを含む HDF5 ファイル ID 番号です。
これは、MMI_OpenH5file コマンドによって返されるファイル ID です。
/F フラグが省略された場合、コマンドは現在アクティブな HDF5 ファイル内の指定されたグループを選択しようとします。

パラメーター

groupID 選択されたグループの ID 番号です。
groupID は、事前に MMI_OpenH5group コマンドによって割り当てられていなければなりません。

詳細

複数の HDF5 ファイルとグループが同時に開かれている可能性があるため、呼び出し元は、既存の適切なグループとファイルを選択していることを確認することが必要です。

参照

MMI_OpenH5group、MMI_CloseH5group、MMI_SelectH5file

MMI_GetEstMaxFrameCount ()

この PC で利用可能な物理メモリの容量から、Igor ウェーブに保存できる、現在アクティブなカメラからの画像の最大数の推定値を返します。

パラメーター

(なし)

戻り値

Igor ウェーブに保存されている、現在、選択されているカメラが取得した各フレームのサイズに対する使用可能な物理メモリのサイズの比率の整数部分（小数点以下切り捨て）[floor] を返します。

詳細

有効な結果を取得するには、カメラデバイスが事前に読み込まれ、初期化され、選択されている必要があります。

例

```
MMC_LoadDevice "myDemoCam", "DemoCamera", "DCam"
MMC_InitializeDevice "myDemoCam"
MMC_SetCameraDevice "myDemoCam"
print MMI_GetEstMaxFrameCount()
    9520 // ウェーブに収まる (demoCamera からの) フレームの最大数

MMC_LoadDevice "myWebCam", "OpenCVgrabber", "OpenCVgrabber"
MMC_InitializeDevice "myWebCam"
MMC_SetCameraDevice "myWebCam"
print MMI_GetEstMaxFrameCount()
    2703 // ウェーブに収まる (webCam からの) フレームの最大数
```

参照

floor、MMC_SetCameraDevice、MMC_GetCameraDevice

MMI_GetCamera /Q /Z /DEF, /L=cameraLabel, /W=winName keyword[=value]

MMI_GetCamera は、Micro-Manager を使って、コンピューターに接続されているカメラ、その現在の設定、およびサポートしているオプションに関する情報を提供します。

これは、v.7 以降で Igor Pro に搭載されていた GetCamera コマンドに代わるものです。

MMI_GetCamera は通常、キーワード/値のペアを含む文字列を、現在のデータフォルダー内の S_info という変数に書き込むことで応答します。

キーワードがサポートされていない場合、応答される値は -1 になります。

availablecameras キーワードを使う場合を除き、MMI_GetCamera への呼び出しでは、アクセスするカメラを指定するために /DEF、/L、または /W フラグのいずれかを使う必要があります。

MMI_GetCamera は、選択されたカメラが露出しているプロパティのリストを提供するために

「properties」キーワードを受け付けます。

（「properties」キーワードおよび詳細セクションを参照してください。）

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsrlt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

/DEF MMI_GetCamera コマンドをコンピューターのデフォルトカメラに適用します。
デフォルトカメラは、開いているビューファインダーウィンドウで表示されている必要はありませんが、コンピューターに接続されている必要があります。

/L=cameraLabel カメラの大文字小文字を区別する Micro-Manager デバイスラベルを含む文字列で、選択されたカメラを指定します。

/W=winName 指定されたグラフウィンドウから選択されたカメラを指定します。
winName は、カメラの映像を表示するグラフウィンドウの名前です（例：graph0）。
カメラの名前または説明ではありません。
MMI は、指定されたウィンドウのノートからカメラの Micro-Manager ラベルを抽出します。

グラフウィンドウが他のグラフ内に埋め込まれている場合、winName は <親グラフ名>#<画像グラフ名> の形式になります。
例えば、summary0#graph0 などです。

キーワード

aperture

キーワード「aperture」はサポートされていません。

availableCameras

セミコロン区切りの文字列 S_CameraNames に含まれる各検出可能なカメラの記述を返します。

/Q フラグを省略すると、コマンドは利用可能なカメラのリストをコマンド ウィンドウの履歴領域に表示します。

「AvailableCameras」キーワードは、他のキーワードとは異なり、/W または /DEF フラグを使ってカメラを指定する必要がなく、他のキーワードと組み合わせて使用できません。

S_CameraNames にリストされているすべてのカメラには、「Adapter : 」と「Name : 」の2つの識別子が、「#」文字で区切られて含まれています。
これらの識別子には、各カメラの Micro-Manager デバイスアダプターとデバイス名の名前が含まれています。例えば、

```
Adapter:DemoCamera#Name:DCam;
```

一部のカメラ（特に OpenCVadapter で実装されたもの）には、各物理デバイスに対応する「Camera : プロパティの値を含む追加の識別子「Property : 」が含まれています。例えば、

```
Adapter:OpenCVgrabber#Name:OpenCVgrabber#Property:Live Streamer CAM 313;
```

colorFilter

キーワード「colorFilter」はサポートされていません。

contrast

キーワード「contrast」はサポートされていません。

customFocusPoint

キーワード「customFocusPoint」はサポートされていません。

denoisingLevel

キーワード「denoisingLevel」はサポートされていません。

description

現在のカメラの説明を返します。

値は、S_info 出力変数にキーワード/値のペアとして返されます：“Description:<value>”。
StringByKey を使って値を取得します。

deviceName

指定されたカメラのデバイス名を返します。
これは通常、16 進数です。

値は、S_info 出力変数にキーワード/値のペアとして返されます：“DeviceName:<value>”。
StringByKey を使って値を取得します。

digitalZoom

キーワード「digitalZoom」はサポートされていません。

exposureCompensation

キーワード「exposureCompensation」はサポートされていません。

focusMode

キーワード「focusMode」はサポートされていません。

isColorFilterSupported=*cf*

キーワード「isColorFilterSupported」はサポートされていません。

isFlashModeSupported=*fm*

キーワード「isFlashModeSupported」はサポートされていません。

isFlashReady

キーワード「isFlashReady」はサポートされていません。

isFocusModeSupported=*fm*

キーワード「isFocusModeSupported」はサポートされていません。

isFocusPointModeSupported=*fpm*

キーワード「isFocusPointModeSupported」はサポートされていません。

isIPAvailable

キーワード「isIPAvailable」はサポートされていません。

isMeteringModeSupported=*mm*

キーワード「isMeteringModeSupported」はサポートされていません。

isWBModeSupported=*wbm*

キーワード「isWBModeSupported」はサポートされていません。

manualWhiteBalance

キーワード「manualWhiteBalance」はサポートされていません。

maximumDigitalZoom

キーワード「maximumDigitalZoom」はサポートされていません。

maximumOpticalZoom

キーワード「maximumOpticalZoom」はサポートされていません。

opticalZoom

キーワード「opticalZoom」はサポートされていません。

orientation

キーワード「orientation」はサポートされていません。

position

キーワード「position」はサポートされていません。

properties

現在のデータフォルダーにある M_cameraProperties というカメラが Micro-Manager に提示するプロパティがリストされている 2 次元テキストウェーブを返します。
各プロパティについて、ウェーブは、プロパティの名前、現在の値、タイプ（整数、文字列、浮動小数点）、許容される最小値と最大値（利用可能な場合）、および各プロパティが読み取り専用（値が固定）であるか、読み取り/書き込み可能（MMC_SetProperty を呼び出すことで値を変更可能）であるかを報告します。

saturation

キーワード「saturation」はサポートされていません。

sharpeningLevel

キーワード「sharpeningLevel」はサポートされていません。

spotMeteringPoint

キーワード「spotMeteringPoint」はサポートされていません。

supportedApertures

キーワード「supportedApertures」はサポートされていません。

supportedISOsensitivities

キーワード「supportedISOsensitivities」はサポートされていません。

supportedShutterSpeeds

キーワード「supportedShutterSpeeds」はサポートされていません。

shutterSpeed

現在のシャッター速度（露出時間）を秒単位で返します。

ドライバーがシャッター速度制御に対応していない場合、-1 を返します。

値は、S_info 出力変数にキーワード/値のペアとして返されます：
"shutterSpeed:<value>"。

NumberByKey を使って値を取得します。

viewfinderPixelFormat

キーワード「viewfinderPixelFormat」はサポートされていません。

viewfinderFrameRateRanges

キーワード「viewfinderFrameRateRanges」はサポートされていません。

viewfinderResolutions

キーワード「viewfinderResolutions」はサポートされていません。

whiteBalanceMode

キーワード「whiteBalanceMode」はサポートされていません。

詳細

「AvailableCameras」キーワードを除き、/DEF フラグを含めるか、カメラの Micro-Manager ラベル（/L フラグを使用）またはカメラ ウィンドウの名前（/W フラグを使用）を指定する必要があります。

Igor の以前の GetCamera コマンドでサポートされていたキーワードのほとんどは、Micro-Manager がサポートするカメラで利用できる機能に対応していません。

そのため、MMI_GetCamera では受け入れられるものの、これらの古いキーワードの多くはサポートされていません。

静寂モードが選択されていない限り、MMI_GetCamera は、サポートされていないキーワードが入力されるたびに、Igor のコマンドウィンドウの履歴セクションに報告します。

Igor の以前の GetCamera コマンドと同様に、MMI_GetCamera は、オペレーティングシステムまたはカメラドライバによって実際にはサポートされていない幅広いオプションを受け入れます。ドライバーが対応する機能をサポートしていない場合、一部の結果は未定義になります。

Micro-Manager で実行されている各デバイスは、デバイスの動作を微調整するためのパラメーター（プロパティ）をいくつか公開しています。

デバイスの動作を調整するプロパティの多くは変更可能ですが、すべてではありません。

properties キーワードで提供される情報と MMC_SetProperty コマンドを組み合わせることで、カメラのプロパティを変更し、その動作をニーズに合わせて調整できます。

出力変数

MMI_GetCamera は、以下の出力変数を設定します：

V_flag

操作が成功した場合は 0、そうでなければゼロではないエラーコードです。

S_cameraNames

「availableCameras」キーワードのみを使って、セミコロン区切りのカメラ説明のリストに設定します。

S_info

利用可能なカメラを除くすべてのキーワードを、セミコロンで区切られたキーワード：値のペアに設定します。

キーワードの結果が複数の可能な値からなる場合、それらの値はコンマで区切られます。

参照

MMI_ModifyCamera、MMI_NewCamera、MMI_GetDevicePropertyList、MMC_SetProperty

MMI_ModifyCamera /Q /Z /L=labelStr keyword=value

MMI_ModifyCamera は、Micro-Manager を使って、指定したカメラの設定を変更します。これは、v.7 以降の Igor Pro に搭載されていた ModifyCamera コマンドに代わるものです。

MMI_ModifyCamera への呼び出しでは、変更対象のカメラを指定するために /L フラグを使う必要があります。

Micro-Manager のカメラへのインターフェイスは、各カメラがサポートする可変のプロパティのセットに依存しているため、MMI_ModifyCamera はすべてのカメラで一貫したインターフェイスを提供することはできません。

従って、MMI_ModifyCamera は、今回廃止された ModifyCamera コマンドで提供されていた多くのキーワードを実装していません。

個々のカメラがサポートするプロパティのセットにばらつきがあるため、MMI_ModifyCamera は個々のカメラが露出するプロパティの大部分へのアクセスを提供しません。

ユーザーが古いコードを更新する際の手間を省き、コンパイルエラーを最小限に抑えるため、MMI_ModifyCamera は ModifyCamera と同じキーワードを受け付けます。

MMI_ModifyCamera コマンドにサポートされていないキーワードを指定して呼び出すと、/Q フラグを使わない限り実行時の警告が発生します。

ユーザーは、各カメラが提供するプロパティのセットを使って、任意のカメラの動作をカスタマイズできます。

カメラとプロパティの詳細については、以下の「詳細」セクションおよび MMC_SetProperty を参照してください。

フラグ

- /Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
- /Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。
- /L=labelStr カメラの大文字小文字を区別する Micro-Manager デバイスラベルを含む文字列で、選択されたカメラを指定します。
- /TO=ms ms はミリ秒単位のタイムアウトです。
画像のキャプチャは、完了までに ms ミリ秒を超える時間がかかった場合、終了されます。

連続キャプチャの場合、タイムアウトは最初のフレームにのみ適用されます。

タイムアウトによりキャプチャが終了した場合、Igor Pro は「データ書き込み中にタイムアウトが発生しました」というエラーを返します。

キーワード

cancelCapture

進行中のキャプチャをキャンセルします。

capture[=dataFolderAndName]

1 フレームをキャプチャして、ウェーブとして保存します。

dataFolderAndName は、ウェーブへのデータフォルダーパス、または現在のデータフォルダー内のウェーブの単純な名前です。

キャプチャを呼び出したときに、ウェーブは存在する場合も存在しない場合もあります。

存在する場合、そのウェーブは上書きされます。

=dataFolderAndName を省略した場合、ModifyCamera は現在のデータフォルダーに M_Frame という名前のウェーブを作成します。

作成されるウェーブは、3 層の RGB ウェーブで、タイプは符号なしバイトです。

キーワードの順序が指定されていないため、他のキーワードを組み合わせずに「capture」キーワードのみを使うことが推奨されます。

ModifyCamera は、Igor に制御を戻す前にキャプチャを実行します。
必要な時間はハードウェアによって異なり、一連のフレームをキャプチャする時の速度の制限を設定します。

customFocusPoint={fx,fy}

キーワード「customFocusPoint」はサポートされていません。

grabSequence={nf,dt}

キーワード「grabSequence」はサポートされていません。

grabToFunction

キーワード「grabToFunction」はサポートされていません。

grabFunctionName=*func*

キーワード「grabFunctionName」はサポートされていません。

setPixelFormat=*pixelFormatStr*

キーワード「setPixelFormat」はサポートされていません。

searchAndLock

キーワード「searchAndLock」はサポートされていません。

setAspectRatio=*{w,h}*

キーワード「setAspectRatio」はサポートされていません。

setAutoAperture

キーワード「setAutoAperture」はサポートされていません。

setAutoISO Sensitivity

キーワード「setAutoISO Sensitivity」はサポートされていません。

setAutoShutterSpeed

キーワード「setAutoShutterSpeed」はサポートされていません。

setCaptureMode=*cm*

キーワード「setCaptureMode」はサポートされていません。

setColorFilter=*cf*

キーワード「setColorFilter」はサポートされていません。

setContrast=*contrast*

キーワード「setContrast」はサポートされていません。

setDenoising=*dn*

キーワード「setDenoising」はサポートされていません。

setExposureComp=*ec*

キーワード「setExposureComp」はサポートされていません。

setExposureMode=*em*

キーワード「setExposureMode」はサポートされていません。

setFlashMode=*fm*

キーワード「setFlashMode」はサポートされていません。

setFocusMode=*fm*

キーワード「setFocusMode」はサポートされていません。

setFocusPointMode=*fpm*

キーワード「setFocusPointMode」はサポートされていません。

setFrameRate=*{min,max}*

キーワード「setFrameRate」はサポートされていません。

setManualAperture=*fn*

キーワード「setManualAperture」はサポートされていません。

setManualIsoSensitivity=*isoInt*

キーワード「setManualIsoSensitivity」はサポートされていません。

`setManualShutterSpeed=ss`

キーワード「setManualShutterSpeed」はサポートされていません。

`setManualWB=temp`

キーワード「setManualWB」はサポートされていません。

`setMeteringMode=mm`

キーワード「setMeteringMode」はサポートされていません。

`setPixelFormat=pf`

キーワード「setPixelFormat」はサポートされていません。

`setResolution={w,h}`

キーワード「setResolution」はサポートされていません。

`setSaturation=st`

キーワード「setSaturation」はサポートされていません。

`setSharpening=sr`

キーワード「setSharpening」はサポートされていません。

`setSpotMeteringPoint={x,y}`

キーワード「setSpotMeteringPoint」はサポートされていません。

`setWBMode=wb`

キーワード「setWBMode」はサポートされていません。

`startPreview`

カメラウィンドウでライブ動画を再生します。

`stopPreview`

カメラウィンドウでのライブ動画を停止します。

`unlock`

キーワード「unlock」はサポートされていません。

`zoomTo={opticalZoom, digitalZoom}`

キーワード「zoomTo」はサポートされていません。

詳細

Igor の以前の `ModifyCamera` コマンドと同様に、`MMI_ModifyCamera` は、オペレーティングシステムまたはカメラドライバで実際にはサポートされていない幅広いオプションを受け入れます。

ほとんどのケースでは、サポートされていないリクエストに対して操作はエラーを返しません。リクエストされた設定がサポートされているかどうかを確実に判断する唯一の方法は、対応するキーワードを使って `MMI_GetCamera` を実行することです。

Igor の以前の `ModifyCamera` コマンドでサポートされていたキーワードのほとんどは、`Micro-Manager` がサポートするカメラで利用できる機能に対応していません。

そのため、`MMI_ModifyCamera` では受け入れられるものの、これらの古いキーワードの多くはサポートされていません。

`Micro-Manager` で動作する各デバイスは、その動作を調整するためのさまざまなパラメーター（プロパティ）を公開しています。

`MMI_GetCamera` コマンドで取得した情報を「properties」キーワードと共に使い、`MMC_SetProperty` コマンドと組み合わせて、カメラのプロパティを変更し、その動作をニーズに合わせて調整できます。

出力変数

MMI_GetCamera は、以下の出力変数を設定します：

V_flag 操作が成功した場合は 0、そうでなければゼロではないエラーコードです。

参照

MMI_NewCamera、MMI_GetCamera、MMI_GetDevicePropertyList、MMC_SetProperty

```
MMI_NewCamera /Q /Z /HIDE=h /HOST=hcSpec /K=k /L=label /S=sequence /N=winName  
/W=(left,top,right,bottom)
```

MMI_NewCamera は、Micro-Manager を使って、サポートされているカメラからのライブビデオを表示する新しいグラフウィンドウを作成します。

これは、v.7 以降の Igor Pro に搭載されていた NewCamera コマンドに代わるものです。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

/HIDE=h ウィンドウを表示 (h=1) または非表示 (h=0、デフォルト) にします。

/HOST=hcSpec 指定された hcSpec で指定されたホストウィンドウまたはサブウィンドウに新しいカメラウィンドウを埋め込みます。
ホストウィンドウまたはサブウィンドウは、コントロールパネルまたはグラフでなければなりません。

hcSpec を使ってサブウィンドウを特定する時は、ウィンドウ階層の構成に関する詳細については「サブウィンドウの構文」を参照してください。

/K=k ユーザーがウィンドウを閉じようとした時のウィンドウの動作を指定します。

k =0: 通常 (ダイアログ表示) (デフォルト)
k =1: ダイアログなしの Kill
k =2: Kill を無効化
k =3: ウィンドウを非表示

/L=label 使用するカメラの Micro-Manager デバイスラベルを指定します。
「label」は大文字小文字を区別する文字列です。

/N=name 作成されたカメラウィンドウに、名前が既に使われている場合は除いて、指定された名前を付けるよう要求します。
既に使われている場合、name0、name1 など、未使用のウィンドウ名が見つかるまで順に試します。

/N を省略した場合、MMI_NewCamera は Camera0 のような名前を使います。

MMI_NewCamera は、選択された名前を S_name に設定します。

/S=n 現在のマシンで利用可能なカメラのシーケンスからカメラ n を選択します。
n は 0 から始まります。
関連するキーワードとして MMI_GetCamera の availableCameras キーワードを参照してください。

/W=(left,top,right,bottom)

ウィンドウの位置を設定します。

座標はトップレベルウィンドウに対してポイント単位で指定されます。

/HOST フラグと組み合わせて使う場合、指定された位置座標は次の 2 つのいずれかの意味を持つ可能性があります：

1. すべての値が 1 未満の場合、座標はホストフレームのサイズに対する分数座標として扱われます。
2. 値が 1 より大きい場合、座標はホストフレームの左上隅を基準点として、ポイント単位で測定された固定位置、またはコントロールパネルホストの場合にはコントロールパネル単位で測定された位置として扱われます。

MMI_NewCamera が存在する場合、コマンドが成功した場合は V_Flag を 0 に設定し、エラーが発生した場合はゼロではないエラーコードを設定します。

詳細

1 台のカメラがコンピューターに接続されている場合は、フラグやキーワードを指定せずに MMI_NewCamera を実行するだけで、ライブビデオを表示することができます。

MoveWindow、SetWindow などを使って、他の Igor ターゲットウィンドウと同じように、結果のグラフウィンドウをコントロールすることができます。

コンピューターに複数のカメラが接続されている場合は、カメラの Micro-Manager デバイスラベルと /L フラグ、または Micro-Manager のロード済みカメラリスト内のカメラの 0 から始まるシーケンス番号と /S フラグを使って、取得するカメラを指定することができます。

(MMI_GetCamera と availableCameras キーワードを参照してください。)

Igor は、作成されたグラフウィンドウを実験とともに保存し、実験を再度開いたときに一致するカメラが見つかった場合、それらをライブモードで開きます。

対応するカメラがない場合でも、Igor はカメラウィンドウを表示しますが、カメラがないことを示すインジケータが表示されます。

MMI_NewCamera が呼び出されるたびに、取得した画像を保持するための固有の名前を持つウェーブを作成します。

このウェーブが誤って変更されるのを防ぐため、取得開始前に、新しく作成されたウェーブをロックします。

対応するグラフウィンドウが閉じられるか、MMI_ReleaseSequenceWave または MMI_Reset が呼び出されるまで、ウェーブはロックされたままになります。

ライブ取得が完了した後、ウェーブを削除または編集する必要がある場合は、変更する前に SetWaveLock を呼び出して、ウェーブのロックを必ず解除してください。

カメラの設定は Experiment と一緒に保存されません。

出力変数

MMI_NewCamera は次の出力変数を設定します：

V_MMIsIt 操作が成功した場合は 0、そうでなければゼロではないエラーコードです。

S_MMImsg 発生したエラーの説明です。コマンドが正常に完了した場合は空白です。

S_Name 新しいカメラウィンドウの名前です。

参照

MMI_GetCamera、MMI_ModifyCamera、MMI_ReleaseSequenceWave、MMI_Reset

MMI_GetXOPversion ()

MMI XOP のバージョン番号を含む Igor 文字列を返します。

例えば、Igor のコマンドラインで次のように入力すると：

```
print MMI_GetXOPversion()
```

次の応答になります：

```
MMI.xop v.1.1.0.3
```

参照

MMC_GetVersionInfo、MMC_GetAPIVersionInfo

Micro-Manager のメソッド

初期化とセットアップ

MMC_LoadDevice /Q /Z deviceLabel, moduleName, deviceName

Micro-Manager に、moduleName および deviceName で指定されたデバイス用の新しいインターフェイスを作成して開き、そのインターフェイスに deviceLabel で指定された ID を割り当てるよう指示します。

特定のモジュール（デバイスアダプター）がサポートするデバイスのリストを取得するには、MMC_GetAvailableDevices を呼び出します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

deviceLevel	このコマンドによって作成されるデバイスのインスタンスを識別するために使われる、大文字小文字を区別するラベルを含む Igor 文字列です。
moduleName	このデバイスに使う Micro-Manager デバイスアダプターの名前の 'root' を含む Igor 文字列です。 (デバイスアダプターに使われるファイル名については、「MMI の使用開始」を参照してください。) 例えば、mmgr_dal_SutterLambda.dll に組み込まれたデバイスアダプターから、名前付きデバイスをロードする場合、moduleName パラメーターは「SutterLambda」になります。
deviceName	アダプターからロードするデバイスの大文字小文字を区別する名前を含む Igor 文字列です。 この名前は、デバイスアダプターの作成者によって設定され、パラメーターはモジュールがサポートするデバイスと完全に一致する必要があります。

参照

Getting Started with MMI、MMC_GetAvailableDevices、MMC_UnloadDevice

MMC_UnloadDevice /Q /Z *deviceLabel*

Micro-Manager に、*deviceLabel* で指定されたデバイスのインターフェイスを解放するよう指示します。指定したデバイスは、MMC_LoadDevice を呼び出して事前に作成しておく必要があります。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

deviceLevel 解放する「デバイス」の大文字小文字を区別するラベルを含む Igor 文字列です。
deviceLabel は、インターフェイスが作成された MMC_LoadDevice の前の呼び出しで使われている必要があります。

参照

Getting Started with MMI、MMC_LoadDevice、MMC_UnloadAllDevices

MMC_UnloadAllDevices /Q /Z

Micro-Manager に、現在ロードされているすべてのデバイスのインターフェイスを削除するよう指示します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

参照

MMC_LoadDevice、MMC_UnloadDevice

MMC_InitializeAllDevices /Q /Z

Micro-Manager に、現在ロードされているすべてのデバイスのインターフェイスを初期化するよう指示します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

参照

MMC_LoadDevice、MMC_InitializeDevice

MMC_Reset /Q /Z

Micro-Manager のコアをリセットし、すべてのデバイスをアンロードして、内部設定をリセットします。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

参照

MMI_Reset、MMC_UnloadAllDevices

MMC_UpdateCoreProperties /Q /Z

Micro-Manager のコアプロパティ（選択したカメラ、シャッター、ホイールなど）を、現在ロードされているデバイスから更新します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

参照

MMC_InitializeAllDevices

MMC_GetCoreErrorText (errCode)

指定されたエラーコードに対応する Micro-Manager エラーテキストを含む文字列を返します。

パラメーター

errCode エラーテキストを返す Micro-Manager エラーコードを示す数値または Igor 数値変数です。

MMC_GetVersionInfo ()

Micro-Manager のコアのバージョン番号を含む文字列を返します。
例えば、Igor のコマンドラインで次のように入力すると：

```
print MMC_GetVersionInfo()
```

次の応答になります：

```
MMCore version 10.4.0
```

参照

MMC_GetAPIVersionInfo、MMI_GetXOPversion

MMC_GetAPIVersionInfo ()

Micro-Manager の Module-API および Device-API のバージョン番号を含む文字列を返します。
例えば、Igor のコマンドラインで次のように入力すると：

```
print MMC_GetAPIVersionInfo()
```

次の応答になります：

```
Device API version 71, Module API version 10
```

参照

MMC_GetVersionInfo、MMI_GetXOPversion

MMC_SaveSysState /Q /Z fileName

Micro-Manager システム（つまり、書き込み可能なプロパティのステータス）の現在の状態を、テキストとして指定したファイルに保存します。
作成されたファイルは、後で MMC_LoadSysState を使って Micro-Manager に状態を再インストールするために使うことができます。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

filename ステータス情報を受信するファイルの名前を指定する Igor 文字列です。

詳細

このコマンドは、同じ名前の既存のファイルを上書きします。
複合パス名で名前を区切るには、必ず二重のバックスラッシュ文字を使用してください。

参照

MMC_LoadSysState、MMC_SaveSysConfig、MMC_LoadSysConfig

MMC_LoadSysState /Q /Z fileName

Micro-Manager システムの状態（つまり、書き込み可能なプロパティのステータス）を、MMC_SaveSysState の以前の呼び出しによって作成された、指定されたファイルから読み込みます。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

filename ロードするステータス情報を含むファイルの名前を指定する Igor 文字列です。

詳細

複合パス名の名前を区切るには、必ず二重のバックスラッシュ文字を使用してください。

参照

MMC_SaveSysState、MMC_SaveSysConfig、MMC_LoadSysConfig

MMC_SaveSysConfig /Q /Z fileName

Micro-Manager に、現在のシステム構成（ロードされているデバイス、そのラベルなど）を Micro-Manager の標準ファイル形式のテキストファイルに保存するよう指示します。
作成されたファイルは、後で MMC_LoadSysConfig を呼び出すことで、構成を再インストールするために使用できます。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

filename 設定を受け取るファイルの名前を指定する Igor 文字列です。

詳細

このコマンドは、同じ名前の既存のファイルを上書きします。
複合パス名の名前を区切るには、必ず二重のバックスラッシュ文字を使用してください。

参照

MMC_LoadSysConfig

MMC_LoadSysConfig /Q /Z fileName

Micro-Manager に、それ以前の MMC_SaveSysConfig の呼び出しで作成されたテキストファイルからシステムの構成をロードするよう指示します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

filename システム構成を含むファイルの名前を指定する Igor 文字列です。

詳細

複合パス名の名前を区切るには、必ず二重のバックスラッシュ文字を使用してください。

参照

MMC_SaveSysConfig

ログ収集とログ管理

MMC_SetPrimaryLogFile /Q /Z fileName logFileName [, truncate]

指定したファイルを Micro-Manager のプライマリ・ログファイルとして設定します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

logFileName Micro-Manager のプライマリ・ログファイルとして使うファイルのフルパス名を指定する Igor 文字列です。
ファイル名が省略された場合、または NULL ("") ファイル名が入力された場合、Micro-Manager のプライマリ・ログファイルは無効になります。

truncate オプションの数値パラメーター（デフォルト値は 0）で、同じパス名を持つ既存のファイルの処理方法を指定します。
truncate が 0 の場合、新しいエントリはファイルの既存の内容に追加されます。
truncate が 0 以外の場合、新しいエントリが追加される前に、ファイルの以前の内容が削除されます。

詳細

複合パス名の名前を区切るには、必ず二重のバックスラッシュ文字を使用してください。

参照

MMC_GetPrimaryLogFile、MMC_LogMessage

MMC_GetPrimaryLogFile ()

Micro-Manager のプライマリ・ログファイルの完全なパス名を含む文字列を返します。

参照

MMC_SetPrimaryLogFile, MMC_LogMessage

MMC_LogMessage /Q /Z message [,debugOnly]

Micro-Manager のプライマリ・ログファイルにメッセージを追加します。

フラグ

- /Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
- /Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

- message ログファイルに追加するメッセージを指定する Igor 文字列です。
- debugOnly オプションの数値パラメーターです（デフォルト値は 0）。
debugOnly が 0 以外の場合、メッセージはデバッグログ記録が有効な場合にのみログファイルに追加されます。
debugOnly が 0 の場合、メッセージは常にログファイルに追加されます。

参照

MMC_GetPrimaryLogFile、MMC_SetPrimaryLogFile、MMC_EnableDebugLog

MMC_EnableDebugLog /Q /Z enableState

Micro-Manager のデバッグメッセージのログ記録を有効または無効にします。

フラグ

- /Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
- /Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

enableState デバッグメッセージの有効状態を設定する数値です。
enableState を 0 に設定すると、デバッグメッセージのログ記録が無効になります。
enableState を 0 以外の値に設定すると、デバッグ メッセージのログ記録が有効になります。

参照

MMC_SetPrimaryLogFile、MMC_LogMessage、MMC_DebugLogEnabled

MMC_DebugLogEnabled ()

パラメーター

(なし)

戻り値

デバッグログの有効/無効を示す数値です。

参照

MMC_SetPrimaryLogFile、MMC_EnableDebugLog

MMC_EnableStdErrLog /Q /Z enableState

Micro-Manager が標準エラー出力にログメッセージを送信するかどうかを有効または無効にします。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

enableState stderr にログメッセージを送信する機能を有効にするための数値設定です。
enableState を 0 に設定すると、stderr へのログメッセージの送信が禁止されます。
enableState を 0 以外の値に設定すると、ログメッセージを stderr に送信できるようになります。

参照

MMC_StdErrLogEnabled

MMC_StdErrLogEnabled ()

パラメーター

(なし)

戻り値

ログ出力が標準エラーに送信されるかどうかを示す数値です。

参照

MMC_SetPrimaryLogFile、MMC_EnableStdErrLog

MMC_StartSecondaryLogFile (logFileName, enableDebug, truncate, synchronous)

Micro-Manager によるログメッセージの追加出力ファイルへのキャプチャを開始します。
作成されたセカンダリ・ログファイルを識別する数値が返されます。

パラメーター

logFileName	セカンダリ・ログファイルのパス名を指定する文字列です。
enableDebug	デバッグメッセージをセカンダリファイルに記録するかどうかを指定する数値です。 プライマリ・ログファイルにデバッグメッセージが記録されるかどうかとは独立しています。 デバッグメッセージをセカンダリ・ログファイルから除外するには、この設定を 0 に設定します。 デバッグメッセージをセカンダリ・ログファイルに含めるには、この設定を 0 以外の値に設定します。
truncate	logFileName で指定されたパス名と一致する既存のファイルをどのように処理するかを指定する数値です。 0 に設定すると、新しいログメッセージは既存のファイルに追加されます。 0 以外の値に設定すると、新しいログメッセージがファイルに書き込まれる前に、既存のファイルがすべて削除されます。
synchronous	セカンダリ・ログファイルへの書き込みが完了する前に、ログ記録関数が呼び出し元に戻るかどうかを指定する数値です。 0 に設定すると、セカンダリ・ログファイルへの書き込みは非同期になります。 0 以外の値に設定すると、セカンダリ・ログファイルへの書き込みは同期処理となり、ログ記録関数が呼び出し元に戻る時点で書き込みが完了していることが保証されます。 Micro-Manager は、同期書き込みは一部のデバッグ状況では役立つが、パフォーマンスが低下するという欠点があると述べています。

戻り値

セカンダリ・ログファイルの作成時に割り当てられる数値識別子です。
この識別子は、MMC_StopSecondaryLogFile を呼び出す時に必要です。

詳細

複合パス名の名前を区切るには、必ず二重のバックスラッシュ文字を使用してください。

参照

MMC_SetPrimaryLogFile、MMC_StopSecondaryLogFile

MMC_StopSecondaryLogFile /Q /Z fileHandle

fileHandle で指定されたセカンダリ・ログファイルへの Micro-Manager のログメッセージの記録を停止します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

fileHandle 選択されたセカンダリ・ログファイルのログ記録を停止する数値識別子です。

参照

MMC_SetPrimaryLogFile、MMC_StartSecondaryLogFile

デバイスのリスト

MMC_GetDeviceAdapterNames /Q /Z [destTextWave]

検出可能なすべてのデバイスアダプターのルート名のリストを取得します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

destTextWave デバイスのリストを受信する、オプションの Igor テキストウェーブです。
既存のウェーブが指定されていない場合、結果は現在のデータフォルダーに
「W_MMCdeviceAdapters」という名前の新しいテキストウェーブに書き込まれます。

詳細

Windows では、Micro-Manager のデバイスアダプターは、次のパターンに一致する名前の .dll ファイルとして実装されていることを思い出してください。

mmgr_dal_<root-name>.dll

MMC_GetDeviceAdapterNames は、Micro-Manager がアクセスできるデバイスアダプターのルート名のリストを返します。

参照

Getting Started with MMI、MMC_GetDeviceName

MMC_GetAvailableDevices /Q /Z /O moduleName [, destTextWave]

指定されたモジュール（デバイスアダプター）から利用可能なデバイスの名前、それぞれのデバイスタイプおよび説明のリストを取得します。

フラグ

- /Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
- /Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsrlt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

moduleName デバイスを一覧表示するモジュール（デバイスアダプター）の「ルート」名です。
Micro-Manager モジュールのルート名は、デバイスアダプターを実装する .dll ファイルの名前に次のように埋め込まれています。

mmgr_dal_<root-name>.dll

destTextWave デバイスリストを受信するオプションの Igor テキストウェーブです。
既存のウェーブが指定されていない場合、結果は現在のデータフォルダーに「W_MMCAvailDevices」という名前の新しいテキストウェーブに書き込まれます。

詳細

MMC_GetDeviceAdapterNames は、Micro-Manager がアクセスできるデバイスアダプターのルート名のリストを返します。

参照

Getting Started with MMI、MMC_GetLoadedDevices、MMC_GetDeviceType、MMC_GetDeviceName、MMC_GetDeviceLibrary

MMC_GetLoadedDevices /Q /Z destTextWave

Micro-Manager のコアにロードされているデバイスの名前の一覧を取得します。

フラグ

- /Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
- /Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsrlt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

destTextWave デバイスリストを受信するオプションの Igor テキストウェーブです。
既存のウェーブが指定されていない場合、結果は現在のデータフォルダーに W_MMCLoadedDevices という名前の新しいテキストウェーブに書き込まれます。

参照

MMC_GetLoadedDevices、MMC_GetDeviceType、MMC_GetDeviceName、MMC_GetDeviceLibrary

一般的なデバイスのコントロール

MMC_GetDeviceName (deviceLabel)

指定されたデバイスラベルに対応するデバイス名を返します。

パラメーター

deviceLabel MMC_LoadDevice のこれまでの呼び出しで割り当てられた、大文字小文字を区別する Micro-Manager デバイスラベルを含む文字列です。

戻り値

ラベルが指定されたデバイスに対応する、大文字小文字を区別する Micro-Manager デバイス名を含む文字列です。

参照

Getting Started with MMI、MMC_GetDeviceLibrary、MMC_GetDeviceType

MMC_GetDeviceLibrary (deviceLabel)

指定されたデバイスラベルのライブラリ（デバイスアダプター）のルート名を返します。

パラメーター

deviceLabel MMC_LoadDevice 関数のこれまでの呼び出しでデバイスに割り当てられたラベルを含む文字列です。

戻り値

デバイスラベルで指定されたデバイスへのインターフェイスを実装するライブラリ（デバイスアダプター）のルート名を含む文字列です。

参照

Getting Started with MMI、MMC_GetDeviceName、MMC_GetDeviceType

MMC_GetDeviceType (deviceLabel)

指定されたデバイスラベルに対応するデバイスタイプを返します。

パラメーター

deviceLabel MMC_LoadDevice 関数のこれまでの呼び出しでデバイスに割り当てられたラベルを含む文字列です。

戻り値

指定されたデバイスラベルが表すデバイスの種類を数値で表した値です。

詳細

返される数値は、以下のとおりデバイスタイプをエンコードしています：

コード デバイスタイプ

0	UnknownType
1	AnyType
2	CameraDevice
3	ShutterDevice
4	StateDevice
5	StageDevice
6	XYStageDevice
7	SerialDevice
8	GenericDevice
9	AutoFocusDevice
10	CoreDevice
11	ImageProcessorDevice
12	SignalIODevice
13	MagnifierDevice
14	SLMDevice
15	HubDevice
16	GalvoDevice

参照

MMC_GetDeviceName、MMC_GetDeviceLibrary

MMC_GetPropertyNumeric (deviceLabel, propertyName)

指定されたデバイスに対して、指定されたプロパティの数値を格納した倍精度数値を返します。

パラメーター

deviceLabel	MMC_LoadDevice 関数のこれまでの呼び出しでデバイスに割り当てられた大文字小文字を区別するラベルを含む文字列です。
propertyName	プロパティの値を検索する時に使う、大文字小文字を区別するプロパティの名前を含む文字列です。

戻り値

指定されたプロパティの値を含む実数値です。

詳細

この関数を使うと、Micro-Manager の数値プロパティタイプ（浮動小数点型および整数型）の値を取得できます。

参照

MMI_GetDevicePropertyList、MMC_GetPropertyString、MMC_SetProperty

MMC_GetPropertyString (deviceLabel, propertyName)

指定されたデバイスに対して、指定されたプロパティの値を含む文字列を返します。

パラメーター

deviceLabel	MMC_LoadDevice 関数のこれまでの呼び出しでデバイスに割り当てられた大文字小文字を区別するラベルを含む文字列です。
-------------	---

propertyName プロパティの値を検索する時に使う、大文字小文字を区別するプロパティの名前を含む文字列です。

戻り値

指定されたプロパティの値を含む文字列です。

参照

MMI_GetDevicePropertyList、MMC_GetPropertyNumeric、MMC_SetProperty

MMC_GetAllowedPropertyValues /Q /Z deviceLabel, propertyName [, destTextWave]

指定されたデバイスラベルのデバイスに対して、指定されたプロパティの許可された文字列値のリストを返します。この機能は、文字列値を使用するプロパティに対してのみ利用可能です。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsrlt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

deviceLabel MMC_LoadDevice 関数のこれまでの呼び出しでデバイスに割り当てられた大文字小文字を区別するラベルを含む文字列です。

propertyName プロパティの値を検索する時に使う、大文字小文字を区別するプロパティの名前を含む文字列です。

destTextWave 許可される値のリストを受け取る、オプションの Igor テキストウェーブです。既存のウェーブが指定されていない場合、結果は現在のデータフォルダーに W_MMCallowedPropertyValues という名前の新しいテキストウェーブに書き込まれます。

参照

MMI_GetDevicePropertyList、MMC_GetPropertyString、MMC_SetProperty

MMC_IsPropertyReadOnly (deviceLabel, propertyName)

指定されたデバイスの指定されたプロパティが読み取り専用であるかどうかを返します。このプロパティには新しい値を割り当てることができません。

パラメーター

deviceLabel MMC_LoadDevice 関数のこれまでの呼び出しでデバイスに割り当てられた大文字小文字を区別するラベルを含む文字列です。

propertyName 大文字小文字を区別する文字列で、チェック対象のプロパティの名前を指定します。

戻り値

数値（0 または 1）で、
1 はプロパティが読み取り専用であることを示し、
0 はプロパティが読み取り専用ではないことを示します。

参照

MMI_GetDevicePropertyList

MMC_IsPropertyPreInit (deviceLabel, propertyName)

指定されたデバイスの名前付きプロパティに値が割り当てられていない場合、そのデバイスが初期化される前に値を割り当てる必要があります。

パラメーター

deviceLabel MMC_LoadDevice 関数のこれまでの呼び出しでデバイスに割り当てられた大文字小文字を区別するラベルを含む文字列です。

propertyName 大文字小文字を区別する文字列で、チェック対象のプロパティの名前を指定します。

戻り値

数値（0 または 1）で、
1 は、デバイスが初期化される前にそのプロパティに値を割り当てる必要があることを示し、
0 は、デバイスが初期化後にそのプロパティに値を割り当てるべきであることを示します。

参照

MMI_GetDevicePropertyList、MMC_SetProperty

MMC_SetProperty /Q /Z /F=floatValue /L=longValue /S=stringValue deviceLabel, propertyName

指定されたデバイスの指定されたプロパティに、指定された値を設定します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

/F=floatValue 数値プロパティの値を設定するために使われ、浮動小数点数（整数でない値）を受け付けるプロパティに適用されます。

/L=longValue 数値プロパティの値を設定するために使われ、整数値のみを受け付けるプロパティに適用されます。

/S=stringValue 文字列値のみを受け付けるプロパティの値を設定するために使われます。

パラメーター

deviceLabel	MMC_LoadDevice 関数のこれまでの呼び出しでデバイスに割り当てられた大文字小文字を区別するラベルを含む文字列です。
propertyName	大文字小文字を区別する文字列で、値が割り当てられるプロパティの名前を指定します。

参照

MMI_GetDevicePropertyList、MMC_GetPropertyNumeric、MMC_GetPropertyString

特定の役割に対する「現在の」デバイスの管理

MMC_GetCameraDevice ()

現在選択されているカメラの Micro-Manager ラベルを含む文字列を返します。

パラメーター

(なし)

戻り値

現在選択されているカメラの大文字小文字を区別するラベルを含む文字列、または選択されていない場合は空の文字列です。

参照

MMC_SetCameraDevice

MMC_SetCameraDevice /Q /Z deviceLabel

指定したデバイスを Micro-Manager のアクティブカメラとして設定します。

フラグ

/Q	「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
/Z	「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。 /Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。 エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

deviceLabel	MMC_LoadDevice 関数のこれまでの呼び出しでデバイスに割り当てられた大文字小文字を区別するラベルを含む文字列です。
-------------	---

参照

MMC_GetCameraDevice、MMC_GetDeviceType

MMC_GetShutterDevice

現在選択されているシャッターの Micro-Manager ラベルを含む文字列を返します。

パラメーター

(なし)

戻り値

現在選択されているシャッターの大文字小文字を区別するラベルを含む文字列、または現在選択されていない場合は空の文字列です。

参照

MMC_SetShutterDevice

MMC_SetShutterDevice /Q /Z deviceLabel

指定したデバイスを Micro-Manager のアクティブシャッターとして設定します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

deviceLabel MMC_LoadDevice 関数のこれまでの呼び出しでデバイスに割り当てられた大文字小文字を区別するラベルを含む文字列です。

参照

MMC_GetShutterDevice、MMC_GetDeviceType

MMC_GetFocusDevice

現在選択されているフォーカスデバイスに対する Micro-Manager ラベルを含む文字列を返します。

パラメーター

(なし)

戻り値

現在選択されているフォーカス デバイスの大文字小文字を区別するラベルを含む文字列、または選択されていない場合は空の文字列です。

参照

MMC_SetFocusDevice

MMC_SetFocusDevice /Q /Z deviceLabel

指定したデバイスを Micro-Manager のアクティブなフォーカスデバイスに設定します。

フラグ

- /Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
- /Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

deviceLabel MMC_LoadDevice 関数のこれまでの呼び出しでデバイスに割り当てられた大文字小文字を区別するラベルを含む文字列です。

参照

MMC_GetFocusDevice、MMC_GetDeviceType

MMC_GetXYStageDevice

現在選択されている XY ステージの Micro-Manager ラベルを含む文字列を返します。

パラメーター

(なし)

戻り値

現在選択されている XY ステージの大文字小文字を区別するラベルを含む文字列、または選択されていない場合は空の文字列です。

参照

MMC_SetXYStageDevice

MMC_SetXYStageDevice /Q /Z deviceLabel

指定したデバイスを Micro-Manager のアクティブな XY ステージデバイスとして設定します。

フラグ

- /Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
- /Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

deviceLabel MMC_LoadDevice 関数のこれまでの呼び出しでデバイスに割り当てられた大文字小文字を区別するラベルを含む文字列です。

参考

MMC_GetXYStageDevice、MMC_GetDeviceType

MMC_GetAutoFocusDevice

現在選択されているオートフォーカスデバイスの Micro-Manager ラベルを含む文字列を返します。

パラメーター

(なし)

戻り値

現在選択されているオートフォーカス デバイスの大文字小文字を区別するラベルを含む文字列、または現在選択されていない場合は空の文字列です。

参照

MMC_SetAutoFocusDevice

MMC_SetAutoFocusDevice /Q /Z deviceLabel

指定したデバイスを Micro-Manager のアクティブなオートフォーカスデバイスとして設定します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

deviceLabel MMC_LoadDevice 関数のこれまでの呼び出しでデバイスに割り当てられた大文字小文字を区別するラベルを含む文字列です。

参照

MMC_GetAutoFocusDevice、MMC_GetDeviceType

MMC_GetSLMDevice

現在選択されている空間光変調器 (SLM) の Micro-Manager ラベルを含む文字列を返します。

パラメーター

(なし)

戻り値

現在選択されている空間光変調器 (SLM) の大文字小文字を区別するラベルを含む文字列、または現在選択されていない場合は空の文字列です。

参照

MMC_SetSLMDevice

MMC_SetSLMDevice /Q /Z deviceLabel

指定したデバイスを Micro-Manager のアクティブな空間光変調器として設定します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

deviceLabel MMC_LoadDevice 関数のこれまでの呼び出しでデバイスに割り当てられた大文字小文字を区別するラベルを含む文字列です。

参照

MMC_GetSLMDevice、MMC_GetDeviceType

MMC_GetGalvoDevice

現在選択されているガルバノメータービームステアリングデバイスの Micro-Manager ラベルを含む文字列を返します。

パラメーター

(なし)

戻り値

現在選択されているガルバノメータービームステアリングデバイスの大文字小文字を区別するラベルを含む文字列、または現在選択されていない場合は空の文字列です。

参照

MMC_SetGalvoDevice

MMC_SetGalvoDevice /Q /Z deviceLabel

指定したデバイスを Micro-Manager のアクティブなガルバノメーターデバイスとして設定します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

deviceLabel MMC_LoadDevice 関数のこれまでの呼び出しでデバイスに割り当てられた大文字小文字を区別するラベルを含む文字列です。

参照

MMC_GetGalvoDevice、MMC_GetDeviceType

画像の取得

MMC_SnapImage /Q /Z

アクティブなカメラから 1 枚の画像を取得します。

Igor ウェーブが 1 つの画像の取得にすでにバインドされている場合、取得された画像も取り出され、バインドされたウェーブに書き込まれます。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

詳細

MMC_SnapImage が、画像を受信するウェーブがバインドされていない状態で呼び出された場合、取得された画像は、MMC_GetImage を呼び出して Igor ウェーブに取得されるか、または MMC_SnapImage が再度呼び出されて上書きされるまで、Micro-Manager の内部画像バッファに保存されます。

参照

MMC_GetImage、MMI_BindImageWave、MMC_SetCameraDevice

MMC_GetImage /Q /Z [destinationWave]

MMC_SnapImage のこれまでの呼び出しで取得した 1 つの画像を取り出し、その画像を Igor ウェーブに転送します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

destinationWave

取得した画像を受け取る数値ウェーブのオプション名です。

imageWave が指定されている場合、そのウェーブ（以前にバインドされた画像ウェーブではなく）が、取得した画像を受け取ります。

詳細

imageWave が指定されていない場合、画像は MMI_BindImageWave の最新の呼び出しで指定されたウェーブに書き込まれます。

imageWave が指定されておらず、単一画像の取得用にウェーブがバインドされていない場合、エラーが発生します。

Igor 画像処理関数が MMI_BindImageProcessor のこれまでの呼び出しで指定されている場合、その関数は、画像がウェーブに書き込まれた後に呼び出されます。

参照

MMC_SnapImage、MMI_BindImageWave、MMI_BindImageProcessor

MMI_GrabImage /Q /Z /ORIG={row,column,layer,chunk} [destinationWave]

1つの画像を取得し、MMC_GetImage よりも細かいコントロールでウェーブに書き込みます。

MMI_GrabImage を使うと、一連の画像を取得して 1つの Igor ウェーブに保存することができます。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

/ORIG={row,column,layer,chunk}

オプションの 4つの整数のセットで、宛先ウェーブ内の画像の原点へのベクトルを指定します。

原点は、取得した画像の最初の（最上端、左端）ピクセルが書き込まれる位置を定義します。

/ORIG フラグを省略した場合、取得した画像は、原点ベクトル {0, 0, 0, 0}、つまり、画像の左上隅のピクセルが 1行目、1列目、1レイヤー、1チャンクとなるようにウェーブに書き込まれます。

パラメーター

destinationWave

取得した画像を受け取る数値ウェーブのオプション名です。

imageWave が指定されている場合、そのウェーブ（以前にバインドされた画像ウェーブではなく）が、取得された画像を受け取ります。

詳細

imageWave が指定されていない場合、画像は MMI_BindImageWave の最新の呼び出しで指定されたウェーブに書き込まれます。

destinationWave が指定されておらず、1つの画像の取得用にウェーブがバインドされていない場合、エラーが発生します。

取得した画像の各行は、宛先ウェーブの列に書き込まれることを思い出してください。

従って、複数の画像を並べて表示するためにウェーブのサイズを決定する場合は、ウェーブの列数を画像の行数の倍数に、ウェーブの行数を画像の列数の倍数に設定してください。

同様に、マルチカラー（RGB など）の画像を扱う場合、各カラー平面は、宛先のウェーブの別々のレイヤーに書き込まれることを思い出してください。

参照

MMC_SnapImage、MMC_GetImage、MMC_GetNumberOfComponents、MMI_BindImageWave、
A Note on Storing and Displaying Images in Igor Waves

MMC_GetNumberOfComponents ()

アクティブなカメラが取得したコンポーネントの数（色チャンネル数）を返します。

パラメーター

(なし)

戻り値

アクティブなカメラによって取得されたコンポーネント（色チャンネル）の数を表す実際の数値です。

詳細

MMC_GetNumberOfComponents は、モノクロカメラの場合に 1 を返し、RGB または RGBA 形式の画像を返すカラーカメラの場合にはそれぞれ 3 または 4 を返します。

また、MMI は現在、RGBA 画像を Igor のウェーブに転送する際に A（アルファ）成分を省略している点にもご注意ください。

MMI がカラー画像をウェーブに転送すると、各色成分はウェーブの別々のレイヤーに書き込まれます。従って、MMI によって取得された RGB カラー画像を格納するウェーブには、少なくとも3つのレイヤーが含まれている必要があります。

参照

A Note on Storing and Displaying Images in Igor Waves、MMI_GrabImage

MMC_GetExposure (cameraLabel)

指定されたカメラの露光時間（ミリ秒単位）を返します。

パラメーター

cameraLabel 指定したカメラの大文字小文字を区別する Micro-Manager ラベルを含む文字列です。パラメーターが空文字列（""）の場合、この関数は現在アクティブなカメラの露出時間を返します。

戻り値

カメラの露出時間をミリ秒単位で表す実際の数値です。

参照

MMC_SetExposure

MMC_SetExposure /Q /Z /L=cameraLabel exposureTime

アクティブなカメラの露光時間を設定します。

フラグ

- /Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
- /Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。
- /L=cameraLabel
露出時間を設定するカメラの大文字小文字を区別する Micro-Manager ラベルを含むオプションの文字列です。
/L フラグが指定されていない場合、指定した露光時間が現在アクティブなカメラに設定されます。

パラメーター

- exposureTime 指定されたカメラに適用される新しい露光時間（ミリ秒単位）を数値で指定する値です。

詳細

カメラは有効な露光時間の範囲がそれぞれ異なります。
ほとんどのカメラでは、MMI_GetDevicePropertyList によって報告されるカメラのプロパティ一覧を確認することで、有効な露光時間の範囲を特定できます。

参照

MMC_GetExposure、MMI_GetDevicePropertyList

MMC_SetROI /Q /Z /L=cameraLabel ROIwave

指定した Igor ウェーブから、指定したカメラの ROI（Region Of Interest）を設定します。

フラグ

- /Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
- /Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。
- /L=cameraLabel
ROI を設定するカメラの大文字小文字を区別する Micro-Manager ラベルを含む文字列です。
/L フラグがない場合、この操作は現在アクティブなカメラに ROI を適用します。

パラメーター

- ROIwave ROI の新しい仕様を保持する、実際の数値ウェーブの名前です。

詳細

ROI は、4 つの整数の列 {x, y, xSize, ySize} として指定されます。
x と y は領域の左上隅のピクセルの水平座標と垂直座標を、xSize と ySize は領域の水平サイズと垂直サイズを表します。

ROIwave が 1 次元ウェーブの場合、ROW 軸に沿った最初の 4 ポイントから仕様が取得されます。
ROIwave が多次元ウェーブの場合、ROW 0 の COLUMN 軸に沿った最初の 4 ポイントから仕様が取得されます。

このスキームは、ROI を定義する柔軟な方法を提供します。
単純な数値のウェーブから、MMC_GetROI または MMC_GetMultiROI の以前の呼び出しで返された ROI の記述を使う場合まで、さまざまな定義が可能です。

参照

MMC_GetROI、MMC_GetMultiROI、A Note on Storing and Displaying Images in Igor Waves

MMC_GetROI /Q /Z /L=cameraLabel [destinationWave]

指定したカメラの対象領域 (ROI) を含む Igor ウェーブを返します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

/L=cameraLabel

ROI を返すカメラの大文字小文字を区別する Micro-Manager ラベルを含む文字列です。
/L フラグを省略した場合、操作は現在アクティブなカメラの ROI を返します。

パラメーター

destinationWave

ROI を受け取る、実際の数値ウェーブのオプション名です。
保存先ウェーブが指定されていない場合、ROI の記述は MMI データフォルダー内の W_camROIs というウェーブに書き込まれます。
その名前の非数値のウェーブが MMI データフォルダーにすでに存在する場合、このコマンドは MMI データフォルダー内に一意の名前を持つウェーブを作成し、そのウェーブに ROI データを書き込みます。

詳細

受信ウェーブは 1 行 4 列で、アクティブな ROI の説明は 1 行目の最初の 4 列 (0 行、0 から 3 列) に配置されます。

ROI の記述は、4 つの整数 x、y、xSize、ySize のシーケンスとして返されます。

このシーケンスは、領域の左上隅のピクセルの x 座標 (水平方向) と y 座標 (垂直方向) を指定し、続いて領域の水平方向と垂直方向のサイズを指定します。

成功すると、ROI 説明を受け取ったウェーブの名前が、MMI データフォルダーの変数 S_MMImresult に書き込まれます。

参照

MMC_SetROI、MMC_GetMultiROI、A Note on Storing and Displaying Images in Igor Waves

MMC_ClearROI /Q /Z

アクティブなカメラの ROI をカメラのフルセンサーに設定し、成功した場合、円形バッファをリセットして、保持していた画像すべてを削除します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

(なし)

参照

MMC_SetROI、MMC_GetROI、MMC_GetMultiROI、
A Note on Storing and Displaying Images in Igor Waves

MMC_IsMultiROISupported ()

アクティブなカメラが複数の ROI を同時に使用できるかどうかを報告します。

パラメーター

(なし)

戻り値

カメラが複数の ROI をサポートしていない場合 : 0
カメラが複数の ROI をサポートしている場合 : 1

参照

MMC_GetMultiROI、MMC_SetMultiROI、MMC_IsMultiROIEEnabled、
A Note on Storing and Displaying Images in Igor Waves

MMC_IsMultiROIEEnabled ()

アクティブなカメラが現在、複数の同時 ROI を使用しているかどうかを報告します。

パラメーター

(なし)

戻り値

カメラが現在複数の ROI を使用していない場合 : 0
カメラが現在複数の ROI を使用している場合 : 1

参照

MMC_GetMultiROI、MMC_SetMultiROI、MMC_IsMultiROISupported、
A Note on Storing and Displaying Images in Igor Waves

MMC_SetMultiROI /Q /Z ROIwave

アクティブなカメラで使用する ROI のセットを定義します。

フラグ

- /Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
- /Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

- ROIwave アクティブなカメラに適用される ROI のセットを表す、実際の数値ウェーブの名前です。

詳細

ROIwave は、少なくとも 2 つの次元（行と列）を有し、少なくとも 1 行と 4 列を有する必要があります。
ROIwave に複数のレイヤーまたはチャンクが含まれている場合、最初のチャンク内の最初のレイヤー（LAYER=0、CHUNK=0）のみが読み込まれます。

各行は、列 0、1、2、3 に格納された 4 の整数のシーケンスとして、1 つの ROI の記述を含むものと解釈されます。

このシーケンスは、x、y、xSize、ySize の順に解釈され、領域の左上隅のピクセルの x 座標（水平方向）と y 座標（垂直方向）を指定し、続いて領域の水平方向と垂直方向のサイズを指定します。

参照

MMC_GetMultiROI、MMC_IsMultiROISupported、MMC_IsMultiROIEnabled、
A Note on Storing and Displaying Images in Igor Waves

MMC_GetMultiROI /Q /Z [ROIwave]

アクティブなカメラで使用中の ROI のセットを表すウェーブを返します。

フラグ

- /Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
- /Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

- ROIwave ROI のリストを受け取る、実際の数値ウェーブのオプション名です。
宛先ウェーブが指定されていない場合、このコマンドは ROI の記述を MMI データフォルダー内の W_camROIs というウェーブに書き込みます。
MMI データフォルダーにその名前而非数値のウェーブがすでに存在する場合、この操作は MMI データフォルダー内に一意の名前を持つウェーブを作成し、そのウェーブに ROI データを書き込みます。

詳細

受信ウェーブは、n 行 4 列で、n はカメラに現在設定されている ROI の数です。

セット内の各 ROI の記述は、連続する行の最初の 4 列（行 0 から n-1 までの列 0 から 3）に配置されます。

各 ROI の記述は、4 つの整数 x、y、xSize、ySize のシーケンスとして保存されます。

このシーケンスは、領域の左上隅のピクセルの x 座標（水平方向）と y 座標（垂直方向）を指定し、その後、各領域の水平方向と垂直方向のサイズが続きます。

成功した場合、この操作は、ROI 記述を受け取るウェーブの名前を MMI データフォルダーの変数 S_MMIResult にも書き込みます。

参照

MMC_SetMultiROI、MMC_IsMultiROISupported、MMC_IsMultiROIEEnabled、
A Note on Storing and Displaying Images in Igor Waves

MMC_GetImageHeight ()

画像バッファの垂直方向のサイズ（つまり行数）をピクセル単位で返します。

パラメーター

（なし）

戻り値

画像バッファ内の 1 フレームの高さ（ピクセル単位）を表す数値です。

参照

MMC_GetImageWidth、A Note on Storing and Displaying Images in Igor Waves

MMC_GetImageWidth ()

画像バッファの水平方向のサイズ（つまり列数）をピクセル単位で返します。

パラメーター

（なし）

戻り値

画像バッファ内の 1 フレームの幅（ピクセル単位）を表す数値です。

参照

MMC_GetImageHeight、A Note on Storing and Displaying Images in Igor Waves

MMC_GetImageBitDepth ()

アクティブなカメラから取得された各画像のピクセルの明るさをエンコードするために使われたビット数を返します。

パラメーター

（なし）

戻り値

アクティブなカメラから返される各ピクセルの明るさをエンコードするために使われるビットの整数値を含む数値です。

詳細

カメラが複数の色平面（例：RGB または RGBA 画像）を返す場合、返される値は各個々の色平面のビット深度を表し、すべての色平面の合計ビット深度ではありません。

参照

MMC_GetNumberOfComponents、

<https://micro-manager.org/apidoc/MMCore/latest/class_c_m_m_core.html>

MMC_SetAutoShutter /Q /Z newState

画像の取得時にアクティブなシャッターの自動開閉を有効または無効にします。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

newState 数値で、0 または 1 のどちらかです。
0 = シャッターの自動動作を無効にします
1 = シャッターの自動動作を有効にします

参照

MMC_SetShutterDevice, MMC_SetShutterOpen

MMC_SetShutterOpen /Q /Z /L=shutterLabel newState

画像の取得時にアクティブなシャッターの自動開閉を有効または無効にします。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

/L=shutterDevice

選択したシャッターを、大文字小文字を区別する Micro-Manager デバイスラベルを含む文字列で指定するオプションのフラグです。
/L フラグを省略した場合、操作は現在アクティブなシャッターの状態を設定します。

パラメーター

newState 数値で、0 または 1 のどちらかです。
0 はシャッターを閉じることを示し（つまり Open は False）、1 はシャッターを開くことを示します（Open は True）。

参照

MMC_SetShutterDevice, MMC_SetAutoShutter

MMC_InitializeCircularBuffer /Q /Z

アクティブなカメラから送信される画像のサイズに基づいて、循環バッファをリセットします。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

詳細

ROI を指定または変更した場合も、Micro-Manager の循環バッファは初期化されます。

参照

MMC_GetCircularBufferSize、MMC_SetCircularBufferSize、MMC_ClearCircularBuffer

MMC_ClearCircularBuffer /Q /Z

Micro-Manager の循環バッファの内容を消去し、存在する画像をすべて消去します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

詳細

ROI が変更された場合、および取得シーケンスが開始された場合にも、Micro-Manager の循環バッファはクリアされます。

参照

MMC_GetCircularBufferSize、MMC_SetCircularBufferSize、MMC_InitializeCircularBuffer、MMC_StartSequenceAcquisition

MMC_GetCircularBufferSize ()

Micro-Manager の循環バッファに割り当てられたメモリの量 (MB 単位) を数値で返します。

パラメーター

(なし)

戻り値

円形バッファのサイズを、MB 単位の整数で表す数値です。

参照

MMC_SetCircularBufferSize、MMC_InitializeCircularBuffer、MMC_ClearCircularBuffer

MMC_SetCircularBufferSize /Q /Z BuffSizeInMB

Micro-Manager の循環バッファのサイズを MB 単位で設定します。

フラグ

- /Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
- /Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

BuffSizeInMB Micro-Manager の循環バッファの新しいサイズを MB 単位で指定する整数値です。

参照

MMC_GetCircularBufferSize、MMC_InitializeCircularBuffer、MMC_ClearCircularBuffer

MMC_IsSequenceRunning ()

Micro-Manager が現在、順次取得を実行しているかどうかを示す数値を返します。

パラメーター

(なし)

戻り値

アクティブなカメラから返される各ピクセルの明るさをエンコードするために使われるビットの整数値を含む数値です。

詳細

カメラが複数の色平面 (例: RGBA) を返す場合、返される値は各色平面のビット深度を表し、すべての色平面の合計ビット深度ではありません。

「この値はガイドラインとしてのみ使用してください。画像バッファに、返されたダイナミックレンジの値のみが含まれることを保証するものではありません。」 [Micro-Manager API より引用。

参照

<https://micro-manager.org/apidoc/MMCore/latest/class_c_m_m_core.html>

MMC_PrepareSequenceAcquisition /Q /Z /L=cameraLabel

カメラを連続撮影の開始に備えて準備します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

/L=cameraLabel

指定したカメラの大文字小文字を区別する Micro-Manager ラベルを含むオプションの文字列です。

/L フラグが省略された場合、コマンドは現在アクティブなカメラを連続取得用に準備します。

参照

MMC_StartSequenceAcquisition、MMC_StopSequenceAcquisition

MMC_StartSequenceAcquisition /Q /Z /N=imageCount /A /O /MD=mode /I=imageInterval

連続取得を開始します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

/N=imageCount

連続で取得する画像の数を指定します。

省略した場合、MMI は連続シーケンスの取得を開始し、新しい画像が取得されるたびに前の画像を上書きします。

/A (/N フラグと併用する場合にのみ有効)

「追加」モードでは、MMI は、これまでに取得した画像を上書きするのではなく、取得した画像をバインドされたストリーミングウェーブに追加します。

MMI は、取得した画像を連続したレイヤー（2 次元画像の場合）として、または連続したチャンク（3 次元画像の場合、例えば RGB や RGBA など）として追加します。

/O 「Overflow」は、循環バッファがいっぱいになった場合に Micro-Manager に取得を停止するよう指示します。

/MD=mode MMI が取得した各画像のメタデータをどのように処理するかを指定します。
 モード 1 と 2 は、MMI_BindMetaDataWave をそれより前に呼び出して、メタデータを受信するテキストウェーブが指定されている場合にのみ有効です。

mode=0 取得した画像のメタデータを保存しません。
 これはデフォルトの設定で、/MD フラグを省略するのと同じです。

mode=1 最も新しく取得した画像のメタデータのみを保存します。

mode=2 各画像を取得する時に、そのメタデータをそのまま保存します。

取得した各画像のメタデータは、バインドされたメタデータウェーブに連続したレイヤーとして追加されます。
 /A および /N フラグも指定されている場合にのみ適用されます。

/I 連続する画像の取得間隔（ミリ秒単位）を指定します。
 省略した場合、間隔は現在のカメラの現在の露出時間に設定されます。

詳細

このコマンドの前に、MMI_BindSequenceWave を呼び出して、着信画像ストリームを受信するウェーブを MMI に通知する必要があります。

参照

MMI_BindSequenceWave、MMI_BindMetaDataWave、MMC_StopSequenceAcquisition、MMC_SetCircularBufferSize

MMC_StopSequenceAcquisition /Q /Z /L=cameraLabel

これまでに開始された連続取得を停止します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
 /Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
 エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

/L=cameraLabel

連続取得を停止するカメラの Micro-Manager ラベルを指定します。
 MMI は現在、一度に 1 台のカメラからの撮影のみに対応しています。

参照

MMC_StartSequenceAcquisition、MMC_IsSequenceRunning

MMI_BindImageWave /Q /Z imageWave

1 つの取得画像を受信するウェーブを定義します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

imageWave 画像を受け取る実際の数値ウェーブです。
フリーウェーブは、画像ウェーブとしてバインドすることはできません。

詳細

imageWave が、現在アクティブなカメラからの 1 フレームを受け入れるのに十分な大きさでない場合、MMI_BindImageWave は、1 フレームを受け入れるようにウェーブのサイズを変更しようとします。
imageWave が 1 フレームより大きい場合、その大きさは変更されません。
同じウェーブは、MMI_BindImageWave と MMI_BindSequenceWave の両方でバインドすることができます。

参照

MMI_ReleaseImageWave、MMC_SnapImage、MMI_GrabImage、MMI_BindSequenceWave

MMI_ReleaseImageWave /Q /Z

1 つの画像の取得用にバインドされていたウェーブを解放します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

詳細

MMI_BindImageWave がまだ呼び出されていない場合、または現在バインドされている画像ウェーブがない場合、MMI_ReleaseImageWave を呼び出してもエラーは発生しません。

参照

MMI_BindImageWave、MMI_BindSequenceWave

MMI_BindSequenceWave /Q /Z /F seqWave

連続取得（ストリーミング）される画像を受け取るウェーブを定義します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。
/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

seqWave 画像シーケンスを受け取る数値ウェーブです。
MMI は、アクティブなカメラから予想される画像のサイズに合わせて、数値タイプとウェーブの大きさをリセットします。
フリーウェーブは、シーケンスウェーブとしてバインドすることはできません。

詳細

同じウェーブを、MMI_BindImageWave と MMI_BindSequenceWave の両方でバインドすることができます。

参照

MMI_ReleaseSequenceWave、MMC_StartSequenceAcquisition、MMC_StopSequenceAcquisition

MMI_ReleaseSequenceWave /Q /Z

画像シーケンスの取得用にバインドされていたウェーブを解放します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

詳細

MMI_BindSequenceWave がまだ呼び出されていない場合、または現在イメージウェーブがバインドされていない場合、MMI_ReleaseSequenceWave を呼び出してもエラーは発生しません。

参照

MMI_BindSequenceWave、MMI_ReleaseImageWave

MMI_BindImageProcessor /Q /Z functionName

MMI が、各（ストリーミングではない）画像の取得後に呼び出す、ユーザー指定の Igor 関数を定義します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

functionName 画像が取得されるたびに呼び出される、ユーザー指定の Igor 機能の名前の文字列です。

詳細

指定された関数は 5 つのパラメーターを受け取り、次に示すように、倍精度実数値を返さなければなりません。

通常、この関数は 0 を返し、エラーを報告する場合にのみ他の値を返します。

```
Function MyImageProcessor (wave imgWave, variable OffRow, variable OffCol, variable  
OffLayer, variable OffChunk)  
    <insert your processing code here>  
    return 0  
End
```

この関数の最初のパラメーターは、数値のウェーブへの参照でなければなりません。

MMI が、指定された関数を呼び出すと、このパラメーターには、新しく取得された画像を受け取ったウェーブへの参照が含まれます。

この関数の 2 番目から 5 番目のパラメーターは、新しく取得した画像が受信ウェーブ内のどこに保存されたかを示します。

受信ウェーブに 1 つの画像しか保存されていない場合、これらのパラメーターはすべて 0 になります。

ただし、MMI_GrabImage を呼び出して複数の画像が 1 つのウェーブに取得されている場合、パラメーターは、新しく取得された画像の原点（左上隅）に対するウェーブ内のベクトルを形成します。

1 つの画像の取得時にバインドされた関数への MMI 呼び出しを停止するには、MMI_ReleaseImageProcessor コマンドを呼び出してください。

参照

MMI_ReleaseImageProcessor、MMC_SnapImage、MMI_GrabImage、MMC_GetImage

MMI_ReleaseImageProcessor /Q /Z

MMI に、各画像の取得後に実行するようにこれまでに定義した、ユーザー指定の Igor 機能呼び出しを停止するように指示します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

詳細

MMI_ReleaseImageProcessor が呼び出された時に、MMI_BindImageProcessor がまだ呼び出されていない場合、または現在バインドされている関数が存在しない場合、エラーは発生しません。

参照

MMI_BindImageProcessor

MMI_BindMetaDataWave /Q /Z metaDataWave

MMI が連続して取得した画像のメタデータを書き込む、ユーザー指定の Igor テキストウェーブを定義します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsrlt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

metaDataWave 連続して取得される画像のメタデータを受け取るテキストウェーブです。
フリーウェーブは、メタデータウェーブとしてバインドすることはできません。

詳細

メタデータ「モード」は、どの画像のメタデータをウェーブに保存するかを指定し、MMC_StartSequenceAcquisition を呼び出す際に /MD フラグで設定されます。
メタデータを保存する場合は、MMC_StartSequenceAcquisition の前に MMI_BindMetaDataWave を呼び出す必要があります。

参照

MMI_ReleaseMetaDataWave、MMC_StartSequenceAcquisition

MMI_ReleaseMetaDataWave /Q /Z

画像シーケンスのメタデータを取得するためにこれまでにバインドされていたウェーブを解放します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsrlt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

詳細

MMI_BindMetaDataWave がまだ呼び出されていない場合、または現在バインドされているメタデータウェーブがない場合、MMI_ReleaseMetaDataWave を呼び出してもエラーは発生しません。

参照

MMI_BindMetaDataWave

MMI_WaitForSequenceCompletion /Q /Z /I=intervalMS /T=timeoutSecs

Micro-Manager が連続取得を行っていないまで待機します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsrlt の非ゼロ値によって報告されます。

エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

/I=intervalMS 連続した取得が進行中かどうかを確認する間の待機時間（ミリ秒単位）を指定します。ユーザー定義のインターバルが定義されていない場合、XOP はデフォルトのインターバル（5 ミリ秒）を使います。

/T=timeOutSecs タイムアウト期間を秒単位で指定します。つまり、操作が連続した取得を停止する連続した取得のチェックを継続する最大期間です。
シーケンシャル取得がタイムアウト期間終了時までには停止していない場合、コマンドはエラーコードを返します。
タイムアウト間隔が指定されていない場合、操作は無限に待機します。以下の警告を参照してください。

詳細

このコマンドは慎重に実行してください。

このコマンドは Igor のメインユーザースレッドで実行され、そのスレッドをロックして Igor のユーザーインターフェイスをフリーズさせる可能性があります。

特に、タイムアウト期間が長く設定されている場合（または設定されていない場合）は注意が必要です。代わりに、独自の Igor プロシージャから MMC_IsSequenceRunning 関数を呼び出すことを検討してください。

参照

MMC_IsSequenceRunning

状態デバイスコントロール

MMC_GetState (deviceLabel)

指定された状態デバイス現在の数値状態を返します。

パラメーター

deviceLabel 指定された状態デバイスに割り当てられた、大文字小文字を区別する Micro-Manager ラベルを含む文字列。

戻り値

指定されたデバイスの動作状態を格納する実際の数値値。

参照

MMC_LoadDevice、MMC_SetState、MMC_SetStateLabel

MMC_DefineStateLabel /Q /Z deviceLabel, stateID, stateLabel

指定されたデバイスの指定された状態に対して、状態ラベル（文字列）を定義します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

deviceLabel 指定された状態デバイスに割り当てられた、大文字小文字を区別する Micro-Manager ラベルを含む文字列です。

newStateID デバイスが選択された動作状態を指定する実際の数値です。

詳細

指定された stateLabel は、より抽象的な数値の stateID で指定された動作状態に対して、便利なまたはユーザーフレンドリーなニックネームを提供します。

参照

MMC_GetState、MMC_SetStateLabel

MMC_SetStateLabel /Q /Z deviceLabel, newStateLabel

指定されたラベル文字列を使って、指定されたデバイスの動作状態を設定します。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

deviceLabel 指定された状態デバイスに割り当てられた、大文字小文字を区別する Micro-Manager ラベルを含む文字列です。

newStateLabel このデバイスで選択された動作状態に割り当てられた状態ラベルを指定する文字列です。

詳細

指定された stateLabel は、通常より抽象的な数値の stateID で指定された動作状態に対して、便利なまたはユーザーフレンドリーなニックネームを提供します。

参照

MMC_GetState、MMC_DefineStateLabel、MMC_SetState

ハブと周辺デバイス

MMC_GetParentLabel (childLabel)

親デバイスのラベルを返すことで、Micro-Manager デバイスの「親」を識別します。

パラメーター

deviceLabel 親デバイスが検索されるデバイスに割り当てられた、大文字小文字を区別する Micro-Manager ラベルを含む文字列です。

戻り値

指定したデバイスの親として以前に設定されたデバイスの、大文字小文字を区別する Micro-Manager ラベルを含む文字列です。

参照

MMC_SetParentLabel

MMC_SetParentLabel /Q /Z deviceLabel, parentLabel

指定されたデバイスの親ハブを設定し、そのデバイスを指定されたハブにリンクします。

フラグ

/Q 「静寂」モードは、MMI に内部エラーを履歴エリアに記録しないように指示します。

/Z 「抑制」モードでは、エラーが発生しても MMI は通常どおり Igor に戻ります。
/Z フラグが使われた場合、エラーの検出はユーザーに委ねられ、数値変数 V_MMIsIt の非ゼロ値によって報告されます。
エラーが発生すると、エラーメッセージがグローバル文字列変数 S_MMImsg に書き込まれます。

パラメーター

deviceLabel 指定された状態デバイスに割り当てられた、大文字小文字を区別する Micro-Manager ラベルを含む文字列です。

parentHubLabel 親デバイスに割り当てられた、大文字小文字を区別する Micro-Manager ラベルを含む文字列です。

参照

MMC_GetParentLabel