NisMacro.Netフレームワーク

概要説明書

# はじめに

## Nisマクロとは/本フレームワークの目的

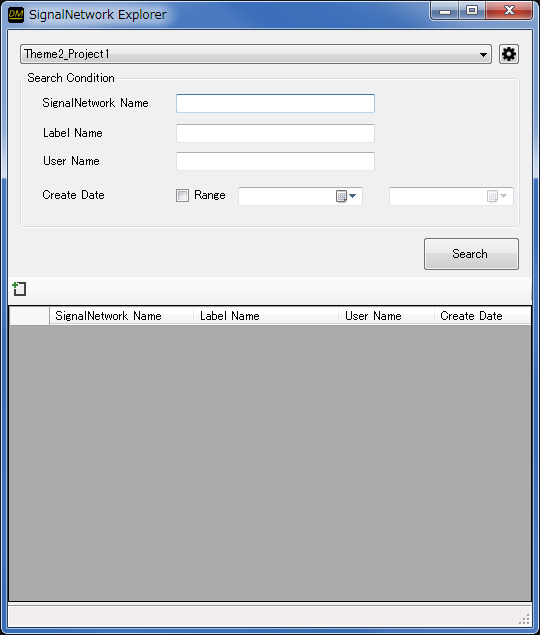
Nis-Elementが持つ機能の一つで、Nis-Elements内のある単位の動作を「マクロ」として実行できるAPIを指す。

「マクロ」機能はNis-Elementsを構成するライブラリの特定の関数を実行することで、外部アプリから実行することが可能である。この機能を用いて、Nis-Elementsから.Netで開発されたアプリケーションを実行し、Nis-Elementを「マクロ」機能経由で制御できる仕組みを実現したのが本フレームワークである。以下に概念図を示す。

Nis-Elements

Macro機能からアプリ起動

マクロ実行



DB

.Netで開発されたアプリ

.Netのライブラリを使用することで実装が容易

装置

etc

## NisMacro.Netフレームワークとは

以下にフレームワークの概要図を示す。

GUI.dll(exe)

(C＃、VB.net等)

NIS-Elements

(nis\_ar.exe)

g5\_new\_ml.dll

g5\_….dll

NisMacro.Net.Core.dll

(C++/CLI)

[起動].mac

NisMacro.Net. Interprocess.dll

(C++/CLI)

NisMacro.Net.Setting.dll

(C＃)

NisMacro.Net.Macro.dll

(C＃)

②アプリ起動の流れ

③マクロ実行の流れ

①DLL参照の流れ

マクロ実行用.mac

NisMacro.Netフレームワーク

NisMacro.Net.Util.dll

(C＃)

NisMacro.Net.Execute.dll

(C＃)

|  |  |
| --- | --- |
| ライブラリ名 | Name Space |
| NisMacro.Net.Core.dll | Nis-Elementsからの起動処理を担う |
| NisMacro.Net.Interprocess.dll | Nis-Elements(g5\_new\_ml.dll)とのマクロ通信を行う  　→コンピュータマインド非公開ソース |
| NisMacro.Net.Macro.dll | マクロオブジェクトとGUIとInterprocessライブラリのインターフェースを担う |
| NisMacro.Net.Execute.dll | NIS-Elementsとのマクロ通信と、GUI側を取り持つ役割を担う。 |
| NisMacro.Net.Setting.dll | 設定値を管理する |
| NisMacro.Net.Util.dll | Nis-Elementsとの連携で使用するユーティリティ機能を持つ |
| NisMacro.Execute,dll | NisMacro.Net.Macro.dllを使用し、NIS-Elementsスレッド上からのマクロ実行を行う。 |

# 処理概要

## ①DLL参照の流れ

各ライブラリは以下の参照設定を行うことが望ましい。

NisMacro.Net.Core

GUI.dll(exe)

NisMacro.Net.Macro

NisMacro.Net.Setting

NisMacro.Net.Util

プロジェクト参照

ライブラリ参照

NisMacro.Net.Interprocess

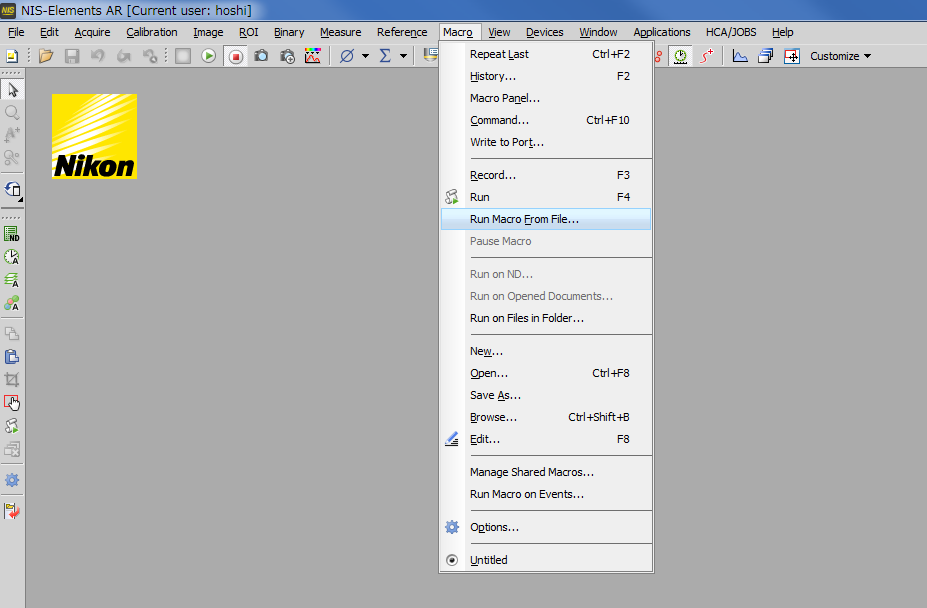
//Dll

Sln管理

NisMacro.Net.Execute

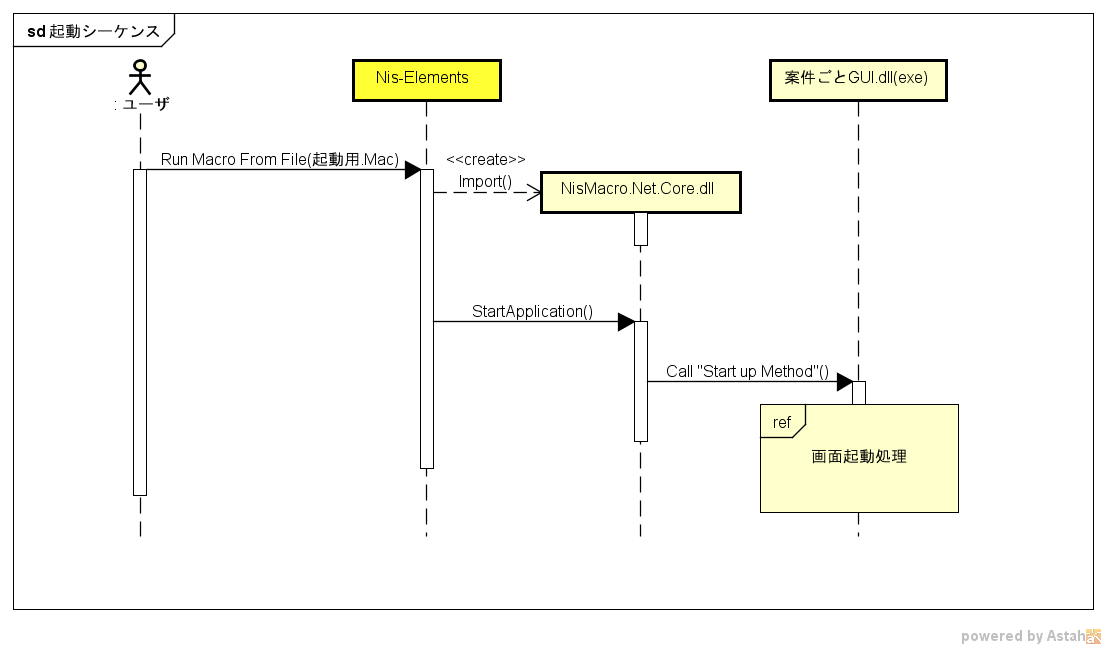
## ②アプリ起動の流れ

本フレームワークはNis-Elementsの「Run Macro From File」機能を用いて起動用のマクロファイルを実行することで起動を行う。また「Run Macro on Events」の「Application Startup」に指定することでNis-Elementsの起動時に自動的にアプリを立ち上げるような設定も可能である。



* シーケンス

以下にアプリの起動を行う際のシーケンスを記載する。



* “起動用.macファイル”の記載例

以下に起動用.macファイルに記載するべきコード例を記載する。

import ("NisMacro.Net.Core.dll"); // ライブラリ名を変更しても特に問題はない

import int StartApplication(); // Module.def ファイルで記載している関数名

int main()

{

StartApplication(); // 2行目でImportした関数を実行する

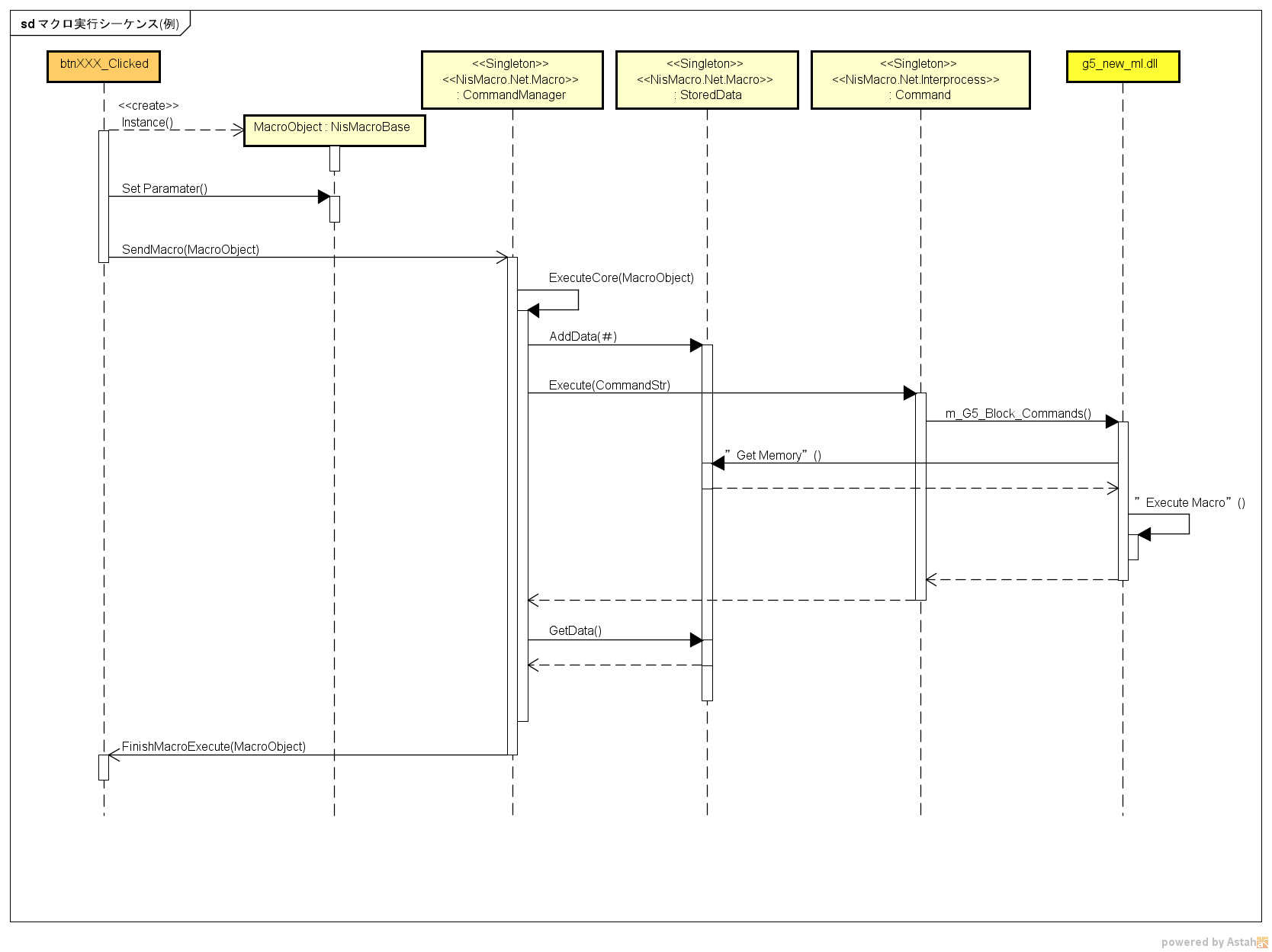
return 0;

}

## ③マクロ実行の流れ

マクロ実行の処理はC#で実装されたマクロオブジェクトを仕様箇所でインスタンスし、NisMacro.Net.MacroプロジェクトのCommandManagerクラスのメソッド「SendMacro（）」を実行することでマクロが実行できる。

以下にマクロ実行時のシーケンス例を示す。マクロオブジェクトの詳細については後述する。



# 開発環境・実行環境

推奨IDEはVisual Studio 2012 以降 (Professional)

※Express版でも実装は実行できるが、一部デバグ機能に制限が生じる場合がある。

また、開発環境がインストールされていないPCで本フレームワークを動作させる場合は以下をインストールする必要がある。

※実行時に「MSVCP110.dll(MSVCP120.dll)」が見つからない旨のメッセージが出た場合、本件が原因

※「MSVCP110d.dll(MSVCP120d.dll)」が見つからないと出た場合は、Debugビルドであることが原因

* Visual Studio2012

[Visual Studio 2012 更新プログラム 4 の Visual C++ 再頒布可能パッケージ](https://www.microsoft.com/ja-jp/download/details.aspx?id=30679)

* Visual Studio2013

[Visual Studio 2013 の Visual C++ 再頒布可能パッケージ](https://www.microsoft.com/ja-jp/download/details.aspx?id=40784)

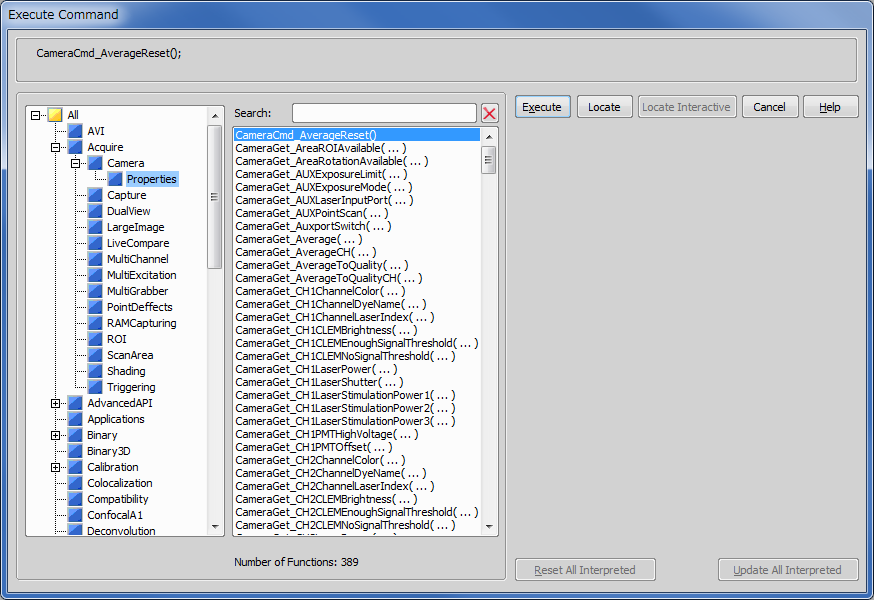
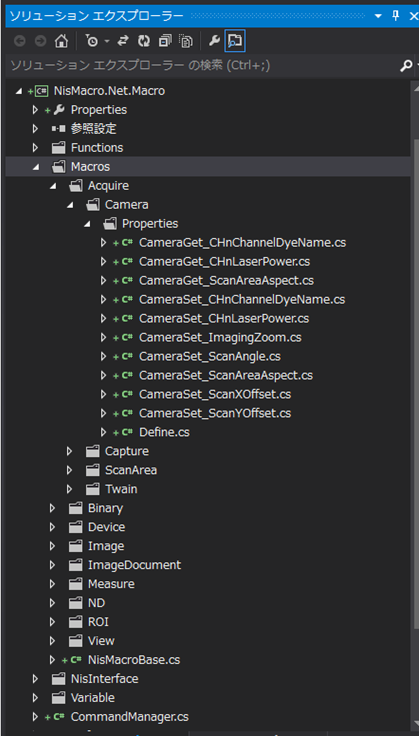
# 開発の手引き

## マクロ追加手順

* 実装ルール

マクロを実装する際は名前空間「NisMacro.Net.Macro.Macros」以下に実装を行う。

管理を容易にするため、実装ルールを以下に示す。

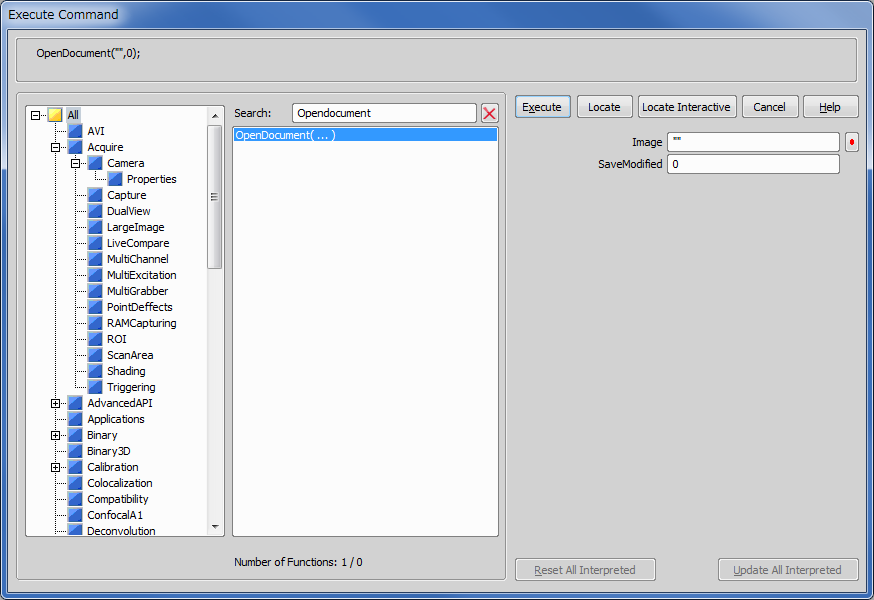
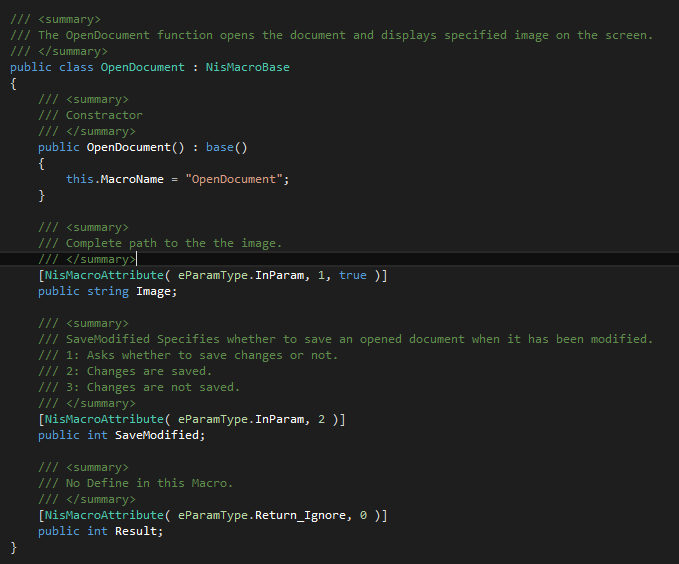


1マクロ = 1ｃｓファイルで実装を行う

名前空間を一致させる

* 実装内容

マクロ「OpenDocument」を例とし、実装する内容を以下に示す。本クラスに書かれている変数・属性を元にNis-Elementsとのデータ送受信を行うため、変数に属性情報を記載すること。



ReturnのIndexは0を指定すること

パラメータの種類（in or out） を指定する

参照渡しが必要な場合はtrueを指定する

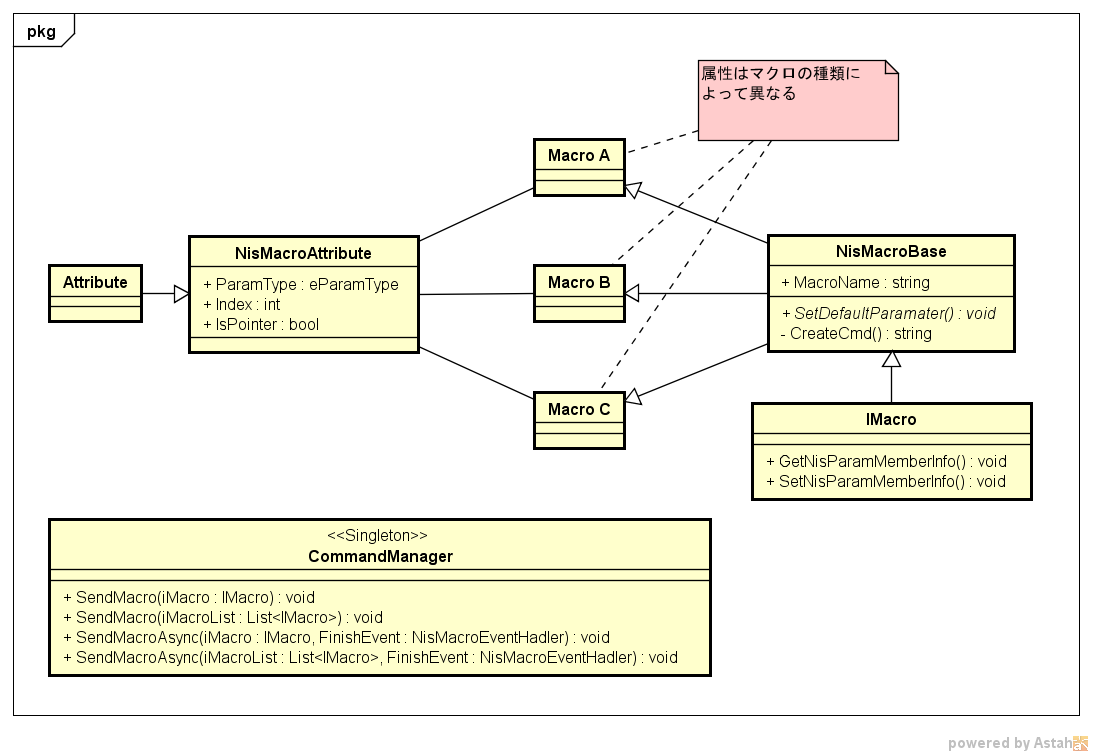
パラメータのIndexを指定する

マクロ名を指定する

変数の型を指定する

* Macroライブラリクラス設計(補足)

以下にマクロ実行関連で使用しているクラス設計を記載する。



# 開発における注意点

・NIS-Elementsの仕様上、マクロを非同期で連続実行することはできません。