

◆UIコントロール機能一覧

通常の記録機能でコントロールがUnknownと記録された場合、UIコントロールをご使用頂くと動作可能となります。※命令や例を参考に、行いたい操作に合うUIコントロール機能をご使用ください。

また、行いたい操作が一覧にない場合は、お問い合わせください。

**【注意】通常の記録機能で取得できるコントロールにUIコントロールを使用しないでください。
誤動作や機能停止の原因となります。**

※UIコントロールを使用しても動作しない場合もございます。

■ウインドウ・コントロール

□状態取得

CheckUIControlExist

(void) CheckUIControlExist(string className, string identifier, bool expectedValue)

コントロールの存在をチェックします。

【命令】

```
Window(“【ウインドウ名】”).CheckUIControlExist(“【コントロール型】”, “【コントロール名】”, “【期待値(true/false)】”);
```

【例】

ウインドウ内にボタン“更新”が存在するかチェックします。

```
PlayLib.Window(“エディタ”).CheckUIControlExist(“UIButton”, “更新”, true);
```

WaitUIControl

コントロールの存在を確認待ちします。

【命令】

```
WaitUIControl(“【コントロール型】”, “【コントロール名】”, “【期待値(true/false)】”);
```

【例】

ウインドウ内にボタン“更新”が存在するかを確認待ちする

```
PlayLib.Window(“ウインドウ名”).WaitUIControl(“UIButton”, “更新”, true);
```

■ボタン

□実行系

Invoke()

必要なメソッドを直接実行します(※推奨)

【命令】

```
UIButton(“【ボタン名】”).Invoke();
```

【例】

ウインドウ上“参照(B)...”ボタンをクリックします。

```
PlayLib.Window(“ファイルの選択”).UIButton(“参照(B)...”).Invoke();□
```

Click()

クリックイベントを発生させます

【命令】

```
UIButton(“【ボタン名】”).Click(【x座標】,【y座標】);
```

【例】

ウィンドウ上“はい”ボタン(の27,13の位置)をクリックします。

```
PlayLib.Window(“ファイルの選択確認”).UIButton(“はい(Y”).Click(27, 13);□
```

□状態取得(状態の確認のみ)

(object) GetProperty(string property)

ボタンの状態(可視)を取得します。

【命令】

```
UIButton(“【ボタン名】”).GetProperty(“【プロパティ名】”);
```

【プロパティ】

プロパティ名	期待値	詳細
Enable	true/false	有効/無効
Visible	true/false	可視

【例】

ボタン名“編集”の可視状態を取得します。

```
bool b = (bool)PlayLib.Window(“エディタ”).UIButton(“編集”).GetProperty(“Visible”);
```

□状態取得(CheckProperty)

(bool) CheckProperty(string property, object expectedValue)

ボタンの状態(可視)をCheckPropertyします。

【命令】

```
UIButton(“【ボタン名】”).CheckProperty(“【プロパティ名】”, “【期待値】”);
```

【プロパティ】

プロパティ名	期待値	詳細
Enable	true/false	有効/無効
Visible	true/false	可視

【例】

ボタン名“編集”の可視状態をCheckPropertyします。

```
PlayLib.Window(“エディタ”).UIButton(“編集”).CheckProperty(“Visible”, true);
```

■チェックボックス

□実行系

Check()

チェックボックスをチェック状態にします。既にチェックオンになっている場合は無視します。

【命令】

```
UICheckBox(“【チェックボックス名】”).Check();
```

【例】

チェックボックス“有効にする(E)”をチェックします。

```
PlayLib.Window(“システム設定”).UICheckBox(“有効にする(E)").Check();□
```

CheckUncheck()

チェックボックスをチェック状態を外します。既にチェックオフになっている場合は無視します。

【命令】

```
UICheckBox(“【チェックボックス名】”).Uncheck();
```

【例】

チェックボックス“有効にする(E)”を外します。

```
PlayLib.Window(“システム設定”).UICheckBox(“有効にする(E)").Uncheck();
```

□状態取得(状態の確認のみ)

(object)GetProperty(string property)

チェックボックスの状態(有効、チェック状態)を取得します。

【命令】

```
UICheckBox(“【チェックボックス名】”).GetProperty(“【プロパティ名】”);
```

【プロパティ】

プロパティ名	戻値	詳細
Enable	true/false	有効/無効
Checked	1/0	チェック状態

【例】

チェックボックス“有効にする(E)”がチェックされているかを取得します。

```
int nNum = (int)PlayLib.Window(“システム設定”).UICheckBox(“有効にする(E)").GetProperty(“Checked”);
```

□状態取得(CheckProperty)

(bool) CheckProperty(string property, object expectedValue)

チェックボックスの状態(有効、チェック状態)をCheckPropertyします。

【命令】

```
UICheckBox("【チェックボックス名】").CheckProperty("【プロパティ名】", "【期待値】");
```

【プロパティ】

プロパティ名	期待値	詳細
Enable	true/false	有効/無効
Checked	1/0	チェック状態

【例】

チェックボックス“有効にする(E)”がチェックされているかをCheckPropertyします。

```
PlayLib.Window("システム設定").UICheckBox("有効にする(E)").CheckProperty("Checked", 1);
```

■ラジオボタン

□実行系

Check()

ラジオボタンをチェック状態にします。既にチェックされている場合は無視します。

【命令】

```
UIRadioButton("【ラジオボタン名】").Check();
```

【例】

ラジオボタン“フォルダを毎回指定する”をチェックします。

```
PlayLib.Window("ライブラリの登録").UIRadioButton("フォルダを毎回指定する").Check();
```

□状態取得(状態の確認のみ)

(object)GetProperty(string property)

ラジオボタンの状態(有効、チェック状態)を取得します。

【命令】

```
UIRadioButton("【ラジオボタン名】").GetProperty("【プロパティ名】");
```

【プロパティ】

プロパティ名	戻値	詳細
Enable	true/false	有効/無効
Checked	1/0	チェック状態

【例】

ラジオボタン“フォルダを毎回指定する”がチェックされているかを取得します。

```
int n = (int)PlayLib.Window("ライブラリの登録").UIRadioButton("フォルダ").GetProperty("Checked");
```

□状態取得(CheckProperty)

(bool) CheckProperty(string property, object expectedValue)

ラジオボタンの状態(有効、チェック状態)をCheckPropertyします。

【命令】

```
UIRadioButton("【ラジオボタン名】").CheckProperty("【プロパティ名】", "【期待値】");
```

【プロパティ】

プロパティ名	期待値	詳細
Enable	true/false	有効/無効
Checked	1/0	チェック状態

【例】

ラジオボタン“フォルダを毎回指定する”がチェックされているかをCheckPropertyします。

```
PlayLib.Window("ライブラリの登録").UIRadioButton("フォルダを毎回指定する").CheckProperty("Checked", 1);
```

■コンボボックス

□実行系

Select()

コンボボックス内のアイテムを選択します。

【命令】

```
UIComboBox("UIComboBox&【何番目】").Select("【アイテム名】");
```

【例】

アドレスバーの"SampleFile"を選択します。

```
PlayLib.Window("エディター").UIComboBox("UIComboBox&1").Select("SampleFile");
```

■エディットボックス

□実行系

SetValue(string text)

エディットボックステキストに値をセットします。

セットにはUIEditコントロールが何番目のコントロールかを調べる必要があります。

【命令】

```
UIEdit("UIEdit&【何番目】").SetValue("【セットする値の内容】");
```

【例】

ウィンドウ上3番目のエディットボックスに"Test.AAA"の文字列をセットします。

```
PlayLib.Window("エディタ").UIEdit("UIEdit&3").SetValue("Test.AAA");
```

Input(string text)

エディットボックステキストに値を入力します。

入力にはUIEditコントロールが何番目のコントロールかを調べる必要があります。

【命令】

```
UIEdit("UIEdit&【何番目】").Input("【入力する値の内容】");
```

【例】

ウィンドウ上3番目のエディットボックスに"Test.AAA"の文字列をセットします。

```
PlayLib.Window("エディタ").UIEdit("UIEdit&3").Input("Test.AAA");
```

□状態取得

Text

エディットボックステキストの中身を取得します

取得にはUIEditコントロールが何番目のコントロールかを調べる必要があります

【命令】

```
UIEdit("UIEdit&【何番目】").Text;
```

【例】

ウィンドウ上3番目のエディットボックスの中身を取得します。

```
string str = PlayLib.Window("エディタ").UIEdit("UIEdit&3").Text;
```

□状態取得

GetProperty(string property)

エディットボックスの状態(有効)を取得します。

【命令】

```
UIEdit("UIEdit&【何番目】").GetProperty("【プロパティ名】");
```

【プロパティ】

プロパティ名	戻値	詳細
Enable	true/false	有効/無効
Visible	true/false	可視
Text	string	文字列

【例】

ウィンドウ上3番目のエディットボックスの中身を取得します。

```
PlayLib.Window("エディタ").UIEdit("UIEdit&3").GetProperty("Text");
```

□状態取得(CheckProperty)

CheckProperty(string property, object expectedValue)

エディットボックスの状態(有効)を取得します。

【命令】

```
UIEdit("UIEdit&【何番目】").CheckProperty("【プロパティ名】", "【期待値】");
```

【プロパティ】

プロパティ名	期待値	詳細
Enable	true/false	有効/無効
Visible	true/false	可視
Text	string	文字列

【例】

ウィンドウ上3番目のエディットボックスの文字列が“編集しました”かどうかCheckPropertyします。

```
PlayLib.Window("エディタ").UIEdit("UIEdit&3").CheckProperty("Text", "編集しました");
```

■スタティック

□状態取得

(object)GetProperty(string property)

エディットボックスの状態(有効)を取得します。

【命令】

```
UIStatic("UIStatic&【何番目】").GetProperty("【プロパティ名】");
```

【プロパティ】

プロパティ名	戻値	詳細
Enable	true/false	有効/無効
Visible	true/false	可視
Text	string	文字列

【例】

ウィンドウ上3番目のエディットボックスの中身を取得します。

```
string str = PlayLib.Window("エディタ").UIStatic("UIStatic&3").GetProperty("Text").ToString();
```

□状態取得(CheckProperty)

CheckProperty(string property, object expectedValue)

スタティクの状態(有効)を取得します。

【命令】

```
UIStatic("UIStatic&【何番目】").CheckProperty("【プロパティ名】", "【期待値】");
```

【プロパティ】

プロパティ名	期待値	詳細
Enable	true/false	有効/無効
Visible	true/false	可視
Text	string	文字列

【例】

```
PlayLib.Window("Microsoft Office*").UIStatic("UIStatic&1").CheckProperty("Text", "プロダクトキーの入力");
```

■ツリービュー

□実行系

Select()

ツリービューのアイテムを選択します。

【命令】

```
UITreeView("UITreeView&【何番目】").Select("【アイテム名】");
```

【例】

ツリービューの“マイライブラリ”を選択します。

```
PlayLib.Window("エディタ").UITreeView("UITreeView&1").Select("マイライブラリ");
```

【補足】

- ・下階層のアイテム名は、先にExpandなどで展開しておく必要があります。
PlayLib.Window("エディタ").UITreeView("UITreeView&1").Expand("マイライブラリ");
- ・下階層のアイテム名は、キャビネット名から“->”で繋げる必要があります。
PlayLib.Window("エディタ").UITreeView("UITreeView&1").Select("マイライブラリ->移動先フォルダ");

Click()

クリックイベントを発生させます

【命令】

```
UITreeView("UITreeView&【何番目】").Click("【アイテム名】");
```

【例】

ツリービューの“マイライブラリ”をクリックします。

```
PlayLib.Window("エディタ").UITreeView("UITreeView&1").Click("マイライブラリ");
```

【補足】

- ・下階層のアイテム名は、キャビネット名から“->”で繋げる必要があります。
PlayLib.Window("エディタ").UITreeView("UITreeView&1").Click("マイライブラリ->移動先フォルダ");

Expand()

アイテムの下階層を展開します。

【命令】

```
UITreeView("UITreeView&【何番目】").Expand("【アイテム名】");
```

【例】

ツリービューの“マイライブラリ”を展開します。

```
PlayLib.Window("エディタ").UITreeView("UITreeView&1").Expand("マイライブラリ");
```

【補足】

下階層のアイテム名は、キャビネット名から“->”で繋げる必要があります。

```
PlayLib.Window("エディタ").UITreeView("UITreeView&1").Expand("マイライブラリ->移動先フォルダ");
```

□状態取得(状態の確認のみ)

(object)GetNodeProperty(string property, string item)

ツリービューのアイテムの状態(存在、選択状態)を取得します。

【命令】

```
UITreeView("UITreeView&【何番目】").GetNodeProperty("【プロパティ名】", "【アイテム名】");
```

【プロパティ】

プロパティ名	戻値	詳細
exists	true/false	存在
selected	true/false	選択状態かどうか

【例】

マイライブラリの下に“Test2”が存在するかどうかを取得します。

```
bool b = (bool)PlayLib.Window("").UITreeView("UITreeView&1").GetNodeProperty("exists", "Test1->Test2");
```

□状態取得(CheckProperty)

CheckNodeProperty(string property, string item, object expectedValue)

ツリービューのアイテムの状態(存在、選択状態)をCheckPropertyします。

【命令】

```
UITreeView("UITreeView&【何番目】").CheckNodeProperty("【プロパティ名】", "【アイテム名】", 【期待値】);
```

【プロパティ】

プロパティ名	期待値	詳細
exists	true/false	存在
selected	true/false	選択状態かどうか

【例】

マイライブラリの下に“Test2”が存在するかどうかをCheckPropertyします。

```
PlayLib.Window("").UITreeView("UITreeView&1").CheckNodeProperty("exists", "Test1->Test2", true);
```

■ツリービューアイテム(ListBox内のアイテムなど)

□実行系

Click()

クリックイベントを発生させます。

【命令】

```
UITreeViewItem("【アイテム名】").Click();
```

【例】

ツリービューアイテムの“新規文書1”をクリックします。

```
PlayLib.Window("エディタ").UITreeViewItem("新規文書1").Click();
```


□状態取得

GetProperty(string property)

ツリービューアイテムの状態(存在、選択状態、チェック状態)を取得します。

【命令】

```
UITreeViewItem("【アイテム名】").GetProperty("【プロパティ名】")
```

【例】

ツリービューアイテム“HTTP 1.1 を使用する”のチェック状態を取得します。

```
int nflag = (int)PlayLib.Window("エディタ").UITreeViewItem("HTTP 1.1 を使用する").GetProperty("Checked");
```

□状態取得

CheckProperty(string property, object expectedValue)

ツリービューアイテムの状態(存在、選択状態、チェック状態)をCheckPropertyします。

【命令】

```
UITreeViewItem("【アイテム名】").CheckProperty("【プロパティ名】",【期待値】)
```

【例】

ツリービューアイテム“HTTP 1.1 を使用する”のチェック状態を確認します。(チェックON=1、チェックOFF=0)

```
PlayLib.Window("エディタ").UITreeViewItem("HTTP 1.1 を使用する").CheckProperty("Checked", 1);
```

■リストビュー

□実行系

Select()

リストビューのアイテムを選択します。

【命令】

```
UIListView("UIListView&【何番目】").Select("【アイテム名】");
```

【例】

リストビューの“Test”を選択します。

```
PlayLib.Window("エディタ").UIListView("UIListView&1").Select("Test");
```

Click()

クリックイベントを発生させます。

【命令】

```
UIListView("UIListView&【何番目】").Click("【アイテム名】");
```

【例】

ツリービューの“Test”をクリックします。

```
PlayLib.Window("エディタ").UIListView("UIListView&1").Click("Test");
```

□状態取得

ItemCount

リストビュー内のアイテム数を取得します。

【命令】

```
UIListView("UIListView&【何番目】").ItemCount;
```

【例】

リストビュー内のアイテム数を取得し、ログに表示します。

```
int result = PlayLib.Window("画面名").UIListView("UIListView&1").ItemCount;  
PlayLib.TestLog("アイテム数は" + result);
```

□状態取得

GetCellProperty(string property, string item, int col)

リストビューのアイテムの状態(存在、選択状態)を取得します。

【命令】

```
UIView("UIView&【何番目】").GetNodeProperty("【プロパティ名】", "【アイテム名】", 【列(0から数える)】);
```

【プロパティ】

プロパティ名	戻値	詳細
exists	true/false	存在
selected	true/false	選択状態かどうか

【例】

リストビューのファイル名に"Test"が存在するかどうかを取得します。

```
bool b = (bool)PlayLib.Window("エディタ").UIView("UIView&1").GetCellProperty("exists", "Test", 0);
```

□状態取得(CheckProperty)

CheckCellProperty(string property, string item, int col, object expectedValue)

リストビューのアイテムの状態(存在、選択状態)を取得します。

【命令】

```
UIView("UIView&1").CheckCellProperty("【プロパティ名】", "【アイテム名】", 【列(0から数える)】, 【期待値】);
```

【プロパティ】

プロパティ名	期待値	詳細
exists	true/false	存在
selected	true/false	選択状態かどうか

【例】

リストビューのファイル名に"Test"が存在するかどうかをCheckPropertyします。

```
PlayLib.Window("エディタ").UIView("UIView").CheckCellProperty("exists", "Test", 0, true);
```

■リストボックス

□実行系

Select()

リストボックス内のアイテムを選択します。

【命令】

```
UIListBox("UIListBox").Select("【アイテム名】");
```

【例】

リストボックス内の"スタンプ1"を選択します。

```
PlayLib.Window("スタンプ一覧").UIListBox("UIListBox").Select("スタンプ1");
```

■タブ

□実行系

Select()

タブを選択します。

【命令1】

```
UITabControl("UITabControl&【何番目】").Select("【アイテム名】");
```

【例1】

タブ“新着文書”を選択します。

```
PlayLib.Window("設定").UITabControl("UITabControl&1").Select("新着文書");
```

【命令2】

```
UITabControl("UITabControl&【何番目】").Select(【何番目】);
```

【例2】

タブ“履歴の一覧”をクリックします。

```
PlayLib.Window("履歴の一覧").UITabControl("UITabControl&1").Select(1);
```

【補足】

UIListBoxコントロールを利用した特殊なタブの場合はClickが効かない場合があります。

Click()

クリックイベントを発生させます。

【命令1】

```
UITabControl("UITabControl&【何番目】").Click("【アイテム名】");
```

【例1】

タブ“新着文書”をクリックします。

```
PlayLib.Window("設定").UITabControl("UITabControl&1").Click("新着文書");
```

【命令2】

```
UITabControl("UITabControl&【何番目】").Click(【何番目】);
```

【例2】

タブ“履歴の一覧”をクリックします。

```
PlayLib.Window("履歴の一覧").UITabControl("UITabControl&1").Click(1);
```

■ツールバー

□実行系

Click()

ツールバーのアイテムをクリックします。

アイテムのクリックにはUIToolbarの中の何番目かを調べる必要があります。

【命令】

```
UIToolbar("UIToolbar&【何番目】").Click(【何番目】);
```

【例】

ツールバーの中の8番目のアイテムをクリックします。

```
PlayLib.Window("エディタ").UIToolbar("UIToolbar&1").Click(8);
```

□状態取得 (CheckProperty)

※CheckProperty系未対応

■イメージ

□実行系

Click()

クリックイベントを発生させます。

【命令】

```
UIImage("UIImage&【何番目】").Click();
```

【例】

3番目のイメージをクリックします。

```
PlayLib.Window("イメージエディタ").UIImage("UIImage&3").Click();
```

■コントロールが何番目かのチェック

□状態取得

GetControlIndexFromPoint

XYポジションを指定し、コントロールが何番目かを取得します。

UIEditやUIStatic等のコントロールが何番目かを取得する為にも使用出来ます。

【命令】

```
GetControlIndexFromPoint("【コントロール名】",【X座標】，【Y座標】);
```

【例】

中央座標(400,240)にあるイメージからコントロール番号を取得し、クリックします。

```
int n = PlayLib.Window("新規文書1 - Document Binder").GetControlIndexFromPoint("UIImage", 400, 240);
if (n > 0)
{
    PlayLib.Window("新規文書1 - Document Binder").UIImage("UIImage&" + n).Click();
}
else
{
    //取得できなかった場合
    PlayLib.TestLogError("コントロールが取得できません。");
}
```

座標(170,35)にあるテキストからStatic番号を取得し、テキストの内容をチェックします。

```
int i = PlayLib.Window("Microsoft Office Professional Plus
2010").GetControlIndexFromPoint("UIStatic",170,35);

//取得した番号のテキストをチェック
PlayLib.Window("Microsoft Office Professional Plus 2010").UIStatic("UIStatic&" + i).CheckProperty("Text",
"プロダクトキーの入力");
```

■ハイパーリンク

□実行系

Click()

クリックイベントを発生させます。

【命令】

```
UIHyperLink("UIHyperLink&【何番目】").Click();
```

【例】

システム画面の“デバイスマネージャー”をクリックします。

```
PlayLib.Window("システム").UIHyperLink("UIHyperLink&3").Click();
```

Invoke()

必要なメソッドを直接実行します(※推奨)

【命令】

```
UIHyperLink("UIHyperLink&【何番目】").Invoke();
```

【例】

システム画面の“デバイスマネージャー”を実行します。

```
PlayLib.Window("システム").UIHyperLink("UIHyperLink&3").Invoke();
```