

I. 本製品の概要

本製品は **TextBox** を継承し、入力時の文字制限などが行えるように拡張したコントロールです。

動作環境について

Microsoft Windows7 以上

開発環境について

Microsoft .NET Framework 4.0 以上

Microsoft VisualStudio 2013

※.NET Framework4.0 以上を使用する事に於いては、VisualStudio のバージョンによりません。

著作権について

このプログラムは入力桁数に制限を設けさせていただいていますが、ご利用される上で問題がなければご自由にお使いください。

なお、著作権は作者である私が保有しています。

本ソフトを運用した結果については、作者は一切責任を負えないのでご了承ください。

アンインストールの方法について

インストーラは作成していませんので、「HcsControls.dll」を削除してください。

この拡張テキストボックスを使用して作成したアプリに関しても、適宜削除をお願い致します。

ご意見、ご要望等がございましたら以下のメール宛てへお願い致します。

hiro.hcs@us.au-hikari.ne.jp

バグ修正履歴

バージョン	修正概要
1.0.7.0	振り仮名を取得する際、漢字確定前に表示されている漢字などの一部を消去すると、振り仮名が正常に取得できない。 その為、上記のような場合は振り仮名を消去する処理を追加。 尚、この修正による操作の変更は無し。

II. プロパティの説明

プロパティ名	型	既定値	機能説明
HCS 共通プロパティ			
DefaultBackColor	System.Drawing.Color	Window	既定の背景色を設定します。
DefaultForeColor	System.Drawing.Color	WindowText	既定の前景色を設定します。
IsPlaySystemSoundWhenError	Bool	True	入力した文字が許可されていない文字の場合、システムサウンドを再生するかを指定します。
IsSelectAllCharacterAtFocused	bool	False	フォーカス取得時に文字を全て選択されている状態にするかを指定します。
IsToBackControlWhenEmpty	bool	False	コントロール内のテキストが、ユーザーの操作により空となった場合、前のコントロールへフォーカスを移動するかを指定します。
IsToNextControlWhenLastDigit	bool	False	コントロール内のテキストが、ユーザー指定の文字数に達した場合、次のコントロールへフォーカスを移動するかを指定します。
NextControlKey	System.Windows.Forms.Keys	None	次のコントロールへ移動する為のキーを指定します。
UpdatedBackColor	System.Drawing.Color	Window	テキスト変更時の背景色を設定します。
UpdatedForeColor	System.Drawing.Color	WindowText	テキスト変更時の前景色を設定します。
HCS 数値入力プロパティ			
Format	String	Empty	数字の書式を指定します。指定する書式文字列は.net の書式文字列に準じます。
IsInputMinus	bool	False	負を表す文字(-)を入力可能とするかを指定します。
IsInputNumeric	bool	True	数字入力とするかを指定します。 このプロパティは true にすると、IsInputCharacter は、自動的に false となります。
NumberOfDigits	HcsTextBoxProperty.NumberPropertyClass	12, 0	入力可能な整数の桁数、小数以下の桁数を指定します。

InitialNumber	HcsCommonProperty. OnOffStringProperty	False, ""	OnOffSwitch を true、SettingString に初期値を設定すると、SettingString と異なる場合は、UpdateBackColor が背景色、UpdateForeColor が前景色として表示されます。
HCS 文字入力プロパティ			
AboutString	HcsTextBoxProperty. StringPropertiesClass	—	入力可能とする文字を指定します。
ControlForFurigana	System. Windows. Forms. Control	Null	フリガナを設定するコントロールを指定します。
IsInputCharacter	bool	False	文字入力とするかを指定します。このプロパティは、このプロパティは true にすると、IsInputNumeric は、自動的に false となります。
KanaType	HcsTextBoxProperty. KanaTypeEnum	Katakana	フリガナのタイプを指定します。既定値は、半角カタカナです。
MaxCharacterDigit	Int	32767	文字列の最大文字数を指定します
InitialString	HcsCommonProperty. OnOffStringProperty	False, ""	OnOffSwitch を true、SettingString に初期値を設定すると、SettingString と異なる場合は、UpdateBackColor が背景色、UpdateForeColor が前景色として表示されます。

Ⅲ. プロパティの詳細

HcsTextBox のプロパティは以下の 3 つのグループに分かれています。

1. HCS 共通プロパティ

以下、2 と 3 に共通で影響を与えるプロパティ群。

変更時の色設定、エラー音等の設定を行います。

2. HCS 数値入力プロパティ

数値入力時に関連するプロパティ群。

正数桁、小数桁の指定、-（マイナス）入力の制御等を設定します。

3. HCS 文字入力プロパティ

文字入力時に関連するプロパティ群。

入力可能となる文字種、桁数等を設定します。

HcsTextBox は、数値入力値、文字入力値を内部で保持している事が特徴です。

その為、数値入力、文字入力を相互に切り替えても入力値が保存されている為、元の入力へ戻すと前に入力した値が表示されます。

例えば、数値入力の時「123」と入力して、文字入力に切り替えて「あいう」と入力します。その後、また数値入力へ切り替えると、先に入力した「123」が表示されます。

次に各プロパティについて説明します。

共通プロパティ

1. **DefaultBackColor** (既定の背景色設定)

通常時の背景色を設定します。標準の **BackColor** と同等です。

2. **DefaultForeColor** (既定の前景色設定)

通常時の前景色を設定します。標準の **ForeColor** と同等です。

3. **IsPlaySystemSoundWhenError** (エラー時 Beep 音再生)

最大桁数以上入力時、入力不可能文字を入力した場合 **Beep** 音を発生させるかを設定します。

コード例：

C#

```
this.hcsTextBox1.IsPlaySystemSoundWhenError = true;
```

Tips：

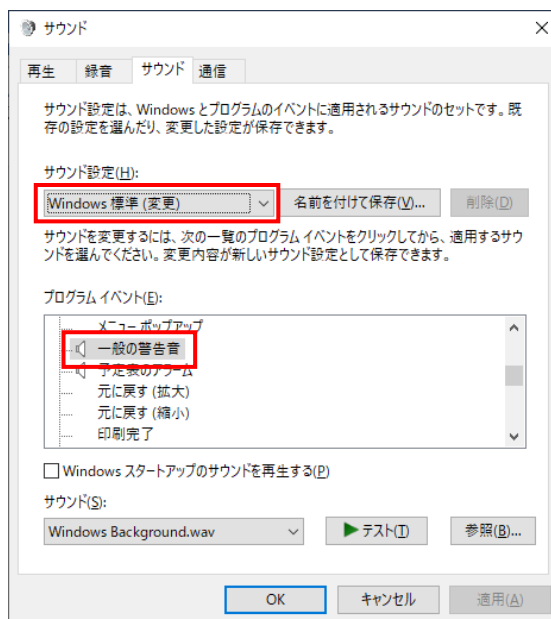
内部でシステム音を再生していますが、音の種類は変更できません。

どうしても変更したい場合は、サウンドコントロールパネルで、下記の通り PC の設定を変更する必

必要があります。

サウンドコントロールパネル

- ① 「サウンド」 タブをクリックする
- ② 「一般の警告音」 をクリックする
- ③ 画面下部の「サウンド」 を変更する



4. IsSelectAllCharacterAtFocused (フォーカス取得時、全テキスト選択)

true に設定すると、フォーカス取得時にすでに入力されているテキストが全て選択されます。

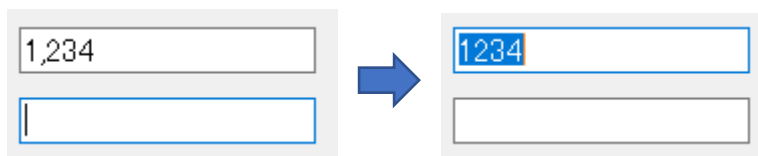
コード例：

C#

```
this.hcsTextBox1.IsSelectAllCharacterAtFocused = true;
```

画面イメージ：

true に設定されていると、フォーカス取得時に全ての文字が選択された状態となります。



5. IsToBackControlWhenEmpty (入力文字列が空になった時、前のコントロールへ移動する)

true に設定すると入力文字列が空になった時、Form で設定した Tab 順の通り、前のコントロールへ移動します。

コード例：

C#

```
this.hcsTextBox1.IsToBackControlWhenEmpty = true;
```

6. IsToNextControlWhenLastDigit (最後の桁を入力した時、次のコントロールへ移動する)

true に設定すると最後の桁入力後、Form で設定した Tab 順の通り、次のコントロールへ移動します。

コード例 :

C#

```
this.hcsTextBox1.IsToNextControlWhenLastDigit = true;
```

7. NextControlKey (指定したキーを入力すると次のコントロールへ移動する)

System.Windows.Forms.Keys で定義されているコントロールを指定すると、指定キーが押されると次のコントロールへ移動します。

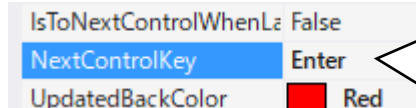
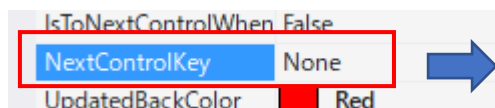
コード例 :

C#

```
this.hcsTextBox1.NextControlKey = Keys.Enter; // Enter キー押下で次のコントロールへ移動
```

Tips :

デザイナーでリスト表示すると、Enter キー等が表示されません。その場合、直接入力を行って下さい。



「Enter」と入力
します。
大文字、小文字の
区別があります。

8. UpdatedBackColor (テキスト変更時の背景色を指定する)

テキスト変更時の背景色を設定します。

テキストが変更されているかどうかの判定は、
数値入力の場合

- InitialNumber の SettingString と Text を比較

文字入力の場合

- InitialString の SettingString と Text を比較

で決定されます。

判定が偽(false)の場合にこのプロパティで設定した色が適用されます。

9. UpdateForeColor (テキスト変更時の前景色を指定する)

UpdateBackColor と同様。ここで設定した色は前景色に適用されます。

数値入力プロパティ

1. Format (書式設定)

フォーカスを持っていない時の書式を設定します。

C#コード例：

```
this.hcsTextBox1.Format = "#, ##0";
```

Tips：

内部で `string.Format` で表示を行っている為、`string.Format` で使用できる書式を設定してください。

2. InitialNumber (初期値)

数値が変更されたかを判定する為の基準値を設定します。



設定：チェックを入れると、入力値と設定した値を比較します。チェックされていない時は、比較しません。

設定値：基準となる数値を入力します。文字列も入力できますが、常に「変更有り」と判定されます。

(前ゼロ、前空白は無視されます)

C#コード例：

```
this.hcsTextBox1.InitialNumber.OnOffSwitch = true;
```

```
this.hcsTextBox1.InitialNumber.SettingString = " 123455";
```

// 上の設定を1度に指定する方法

```
this.hcsTextBox1.InitialNumber = new HcsControls.HcsCommonProperty.OnOffStringProperty(true, " 12345");
```

3. IsInputMinus (- の入力を可能とします)

このプロパティを `true` にすると - (マイナス) の入力が可能となります。

C#コード例：

```
this.hcsTextBox1.IsInputMinus = true;
```

Tips：

`true` を入力するとカーソルがどの位置にあっても先頭へ - が入力され、もう一度 - を押すと - が削除されます。



※カーソルがどの位置にあっても、先頭へ - が追加されます。入力位置は変わりません。

4. IsInputNumeric(数値の入力を行う)

このプロパティを true へ設定すると数値のみの入力設定となります。

※このプロパティを true に設定すると、自動的に IsInputCharacter は false となります。

C#

```
this.hcsTextBox1.IsInputNumeric = true;
```

5. NumberOfDigits(入力を許可する桁数を指定)

入力する桁数を、正数桁、小数桁を指定します。

コード例：

C#

```
// 一括指定する場合(正数 4 桁、小数 2 桁の入力を指定)
```

```
this.hcsTextBox1.NumberOfDigits = new HcsControls.HcsTextBoxProperty.NumberPropertyClass(4, 2);
```

```
// 個別に指定する場合(正数 4 桁、小数 2 桁の入力を指定)
```

```
this.hcsTextBox1.NumberOfDigits.IntegerDigit = 4;
```

```
this.hcsTextBox1.NumberOfDigits.DecimalDigit = 2;
```

Tips：

両方に 0 を指定した時は、正数、小数ともに入力可能となり正数桁と小数桁を合わせて 32,767 桁となります。

小数桁を入力させないようにする為には、正数桁に 0 以外を指定して、小数桁を 0 とする必要があります。

1. AboutString(入力文字列の指定)

入力可能文字列を指定します。

設定値	IsHalf	半角文字の入力を許可します
	IsFull	全角文字の入力を許可します
	IsAlphabet	英字の入力を許可します
	IsNumber	数字の入力を許可します
	IsKatakana	カタカナの入力を許可します
	IsHiragana	ひらがなの入力を許可します
	AllowString	追加で許可したい文字列を正規表現で指定します
	NotAllowString	拒否したい文字列を正規表現で指定します

C#

```
// 以下の例では、全角と半角の英数字を許可しています
var prop = new HcsControls.HcsTextBoxProperty.StringPropertiesClass();
// IsFull(全角)、IsHalf(半角)の両方を入力可能とする
prop.IsFull = true;
prop.IsHalf = true;
prop.IsAlphabet = true;
prop.IsNumber = true;
this.hcsTextBox1.AboutString = prop;
```

デザイナーによる指定(上記と同一の設定)

The screenshot shows the HcsTextBoxProperty designer interface. It contains the following elements:

- A group box with three radio buttons:
 - ☐ 半角
 - ☐ 全角
 - ☒ 半角 & 全角
- Four checked checkboxes:
 - ☒ 英字
 - ☒ 数字
 - ☐ カタカナ
 - ☐ ひらがな
- A label "許可文字(正規表現)" followed by an empty text input field.
- A label "許可しない文字(正規表現)" followed by an empty text input field.

許可文字、許可しない文字について

例えば、一般の車両番号の入力に関して設定する場合は、下記のように設定します。

The screenshot shows a settings window for input validation. It includes radio buttons for '半角' (Half-width), '全角' (Full-width), and '半角 & 全角' (Half-width & Full-width), with the last one selected. Below are checkboxes for '英字' (Alphabet), '数字' (Numbers), 'カタカナ' (Katakana), and 'ひらがな' (Hiragana), with '数字' checked. There are two text input fields: '許可文字(正規表現)' (Allowed characters (Regular expression)) containing 'あ-ん¥-' and '許可しない文字(正規表現)' (Not allowed characters (Regular expression)) containing 'われ'. Callout boxes provide additional context: '半角、全角を許容 ※半角指定でも OK' points to the radio buttons; '数字を許容' points to the checked '数字' checkbox; 'あ〜んまでを許容、更に -(マイナス) を許容' points to the '許可文字' field; and 'レンタカーを表す、'わ' と'れ'は入力不可' points to the '許可しない文字' field.

※実際は、もっと細かい制御が必要となりますので、プログラムで制御する必要があります。また、上部で数字のみを設定しても許可文字で指定されている文字は入力可能となり、正規表現に関しては 1 文字単位でチェックが入ります。文字列単位での指定はできません。

2. ControlForFurigana (入力された文字列の振り仮名を設定するコントロール)

Form に配置されているコントロールを指定すると、指定したコントロールの Text プロパティへ振り仮名を設定します。

C#

```
// textBox1 を振り仮名の設定先へ指定します  
this.hcsTextBox1.ControlForFurigana = this.textBox1;
```

3. InitialString (初期値)

文字列が変更されたかを判定する為の基準値を設定します。

The screenshot shows a small dialog with a checked checkbox labeled '設定' (Setting). Below it is a text input field containing 'あああ' (aaa), which is highlighted by a red rectangle. To the right of the text field is the label '設定値' (Setting value).

設定：チェックを入れると、入力値と設定した値を比較します。チェックされていない時は、比較しません。
設定値：基準となる文字列を入力します。許可していない文字も入力できますが、常に「変更有り」と判定されます。（前空白も判定に含まれます）

C#コード例：

```
this.hcsTextBox1.InitialString.OnOffSwitch = true;
```

```
this.hcsTextBox1.InitialString.SettingString = "あああ";
```

// 上の設定を同時に指定

```
this.hcsTextBox1.InitialNumber = new HcsControls.HcsCommonProperty.OnOffStringProperty(true, "あああ");
```

4. IsInputCharacter (文字の入力を行う)

このプロパティを true へ設定すると文字の入力設定となります。

※このプロパティを true に設定すると、自動的に IsInputNumeric は false となります。

C#

```
this.hcsTextBox1.IsInputCharacter = true;
```

5. KanaType (振り仮名の形式)

以下の値を設定可能です。

設定名	内容
Katakana	振り仮名として、半角カタカナを選択
KatakanaWide	振り仮名として、全角カタカナを選択
Hiragana	振り仮名として、ひらがなを選択

ControlForFurigana で指定したコントロールへ、どのタイプの振り仮名を設定するかを指定します。

C#

// 以下の例は、指定コントロールへひらがなのふりがなが設定されます

```
this.hcsTextBox1.KanaType = HcsControls.HcsTextBox.KanaTypeEnum.Hiragana;
```

6. MaxCharacterDigit (文字の最大桁数)

文字入力時に、入力文字数を何桁とするかを指定します。

※全角、半角ともに1文字とカウントします。