

WireLessFlexBrowser

Softogo.Print – Manuel de l'utilisateur

Softogo.Print – Manuel de l'utilisateur.doc

20040915 Version 1.0	
20100811 Version 1.1	Caractères hexadécimales ajoutées

Utilisation de Softogo.Print

Declaration

Pour déclarer l'objet Softogo.Print dans une page HTML la ligne de code suivante doit être incluse dans le BODY.

```
<OBJECT
  ID="PrintObj" CLASSID="CLSID:5C93AC05-C238-49C1-8CB4-E9D8A9FA242C">
</OBJECT>
```

Methodes

L'objet compte les methodes suivantes.

AddData ("Add data to the buffer.")
 AddLine ("Add a line, data + end-line, to the buffer.")
 Print (copies)
 Clear ()
 PrintFromFile ("\\print\\filetoprint.txt", copies)
 GetLastError()
 GetLastErrorDes()

Pour chaque fonction qui ajoute un texte on peut utiliser la sequence d'escape "XX" pour imprimer le caractère hexadecimale XX. (du \00, \01, ... au \FF).

AddData ("Add data to the buffer.")

Ajouter texte au buffer **sans** un retour à la ligne. La sequence d'escape "\n" peut être utilisée, au milieu ou à la fin, pour indiquer des retour à la ligne.

Exemples

Commande	Output
AddData ("Data1")	Data1←
AddData ("Data1\n")	Data1␣ ←
AddData ("Data1\nData2")	Data1␣ Data2←
AddData ("Data1\nData2\n")	Data1␣ Data2␣ ←

AddLine ("Add a line, data + end-line, to the buffer.")

Ajouter texte au buffer **avec** un retour à la ligne. La sequence d'escape "\n" peut être utilisée, au milieu ou à la fin, pour indiquer d'autres retours à la ligne.

Exemples:

Commande	Output
AddLine ("Data1")	Data1 ↵ ←
AddLine ("Data1\nData2")	Data1 ↵ Data2 ↵ ←

Print (copies)

Imprimer le buffer le nombre de fois indiquées avec le parametre.

Clear ()

Effacer le buffer.

PrintFromFile ("\\print\\filetoprint.txt", copies)

Imprimer le fichier le nombre de fois indiquées avec le parametre.

GetLastError()

Retourner le code du dernier erreur. Zéro indique que la dernière operation n'a pas provoqué d'erreur.

GetLastErrorDes()

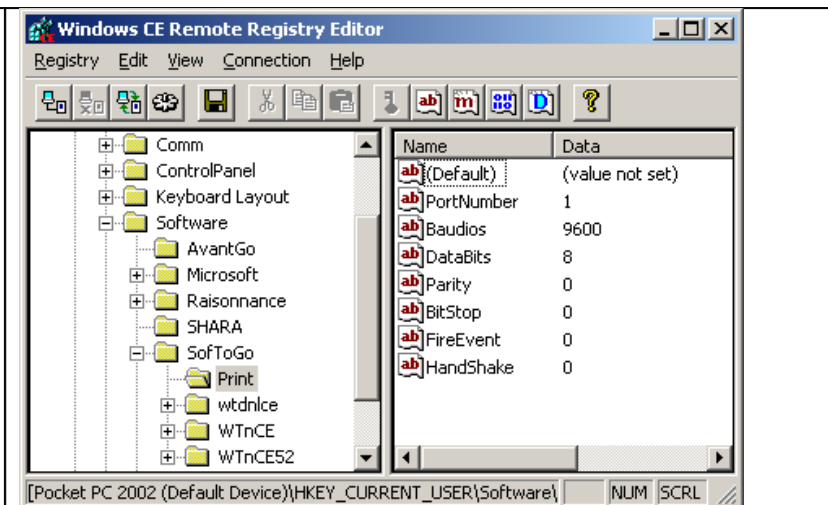
Retourner la description du dernier erreur. Une chaîne vide indique que la dernière operation n'a pas provoqué d'erreur.

Configuration du port de communications

L'objet Softogo.Print initialise le port serie de communication selon les informations prises de la registry du Pocket PC.

```

HKEY_CURRENT_USER
  Software
    Softogo
      Print
        PortNumber = "1"
        Baudios = "9600"
        DataBits = "8"
        BitStop = "0"
        Parity = "0"
        FireEvent = "0"
        HandShake = "0"
          
```



PortNumber

Numero de port utilisé pour la communication. Par exemple 1.

Bauds

Le débit des données en bauds. Les valeurs acceptées sont:

110
300
600
1200
2400
4800
9600
14400
19200
38400
56000
57600
115200
128000
256000

HandShake

Le protocole interne de communications utilisé pour la transmission à niveau de hardware.

0	None
1	Hardware
2	Software

DataBits

Nombre de bits pour la transmission. Les valeurs acceptées sont: 7, 8.

Parity

Type de Parité. Les valeurs possibles sont:

0	None
1	Odd
2	Even
3	Mark
4	Space

BitStop

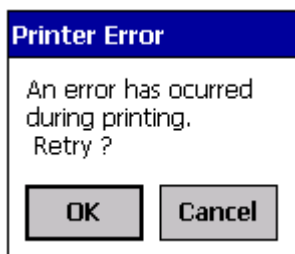
Nombre de bits d'arrêt.

0	1 bit
1	1.5 bits
2	2 bits

FireEvent

Action à suivre après un erreur.

1	Lance un evennement. No reessaie pas.
0	Montre un message d'erreur et permet de reessaiier.

**Exemples complets****Impression simple**

L'exemple suivant imprime trois lignes consecutives en appuyant sur le button Imprimer.

```
<HTML>
  <HEAD>
    <TITLE>ATL 3.0 test page for object Print</TITLE>
  </HEAD>

  <SCRIPT ID=clientEventHandlersJS LANGUAGE=javascript>
```

```

function Imprimer()
{
    PrintObj.AddData("Texte 1, ligne 1\nTexte 2, ligne 2\n");
    PrintObj.AddLine("Texte 3, ligne 3");
    PrintObj.Print(1);
    PrintObj.Clear();
}
</SCRIPT>

<BODY>
<OBJECT
    ID="PrintObj"
    CLASSID="CLSID:5C93AC05-C238-49C1-8CB4-E9D8A9FA242C">
</OBJECT>
<input type="button" onclick="Imprimer()"
    value="Imprimer" id=button1 name=button1>
</BODY>
</HTML>

```

Impression simple à partir d'un fichier

L'exemple suivant imprime 2 copies du contenu du fichier text.txt qui se trouve dans le repertoire toprint.

```

<HTML>
<HEAD>
<TITLE>ATL 3.0 test page for object Print</TITLE>
</HEAD>

<SCRIPT ID=clientEventHandlersJS LANGUAGE=javascript>
function Imprimer()
{
    PrintObj.PrintFromFile("\\toprint\\text.txt", 2);
}
</SCRIPT>

<BODY>
<OBJECT
    ID="PrintObj"
    CLASSID="CLSID:5C93AC05-C238-49C1-8CB4-E9D8A9FA242C">
</OBJECT>
<input type="button" onclick="Imprimer()"
    value="Imprimir" id=button1 name=button1>
</BODY>
</HTML>

```

Capture d'un evenement face à un erreur

Pour generer un erreur connu, on demande à ActiveX d'imprimer quand le port de communication n'est pas disponible. Puis, on établie dans la registry que, face à un erreur, un evenement soit lancé. Cela est fait avec le parametre FireEvent="1".

Quand l'erreur est generé, ActiveX lance un evenement PrintError, qui est capturé par le script au dessous de la declaration de l'objet PrintObj.

Ce script montre le code d'erreur et sa description.

```

<HTML>
<HEAD>
<TITLE>ATL 3.0 test page for object Print</TITLE>
</HEAD>
<SCRIPT ID=clientEventHandlersJS LANGUAGE=javascript>

```

```

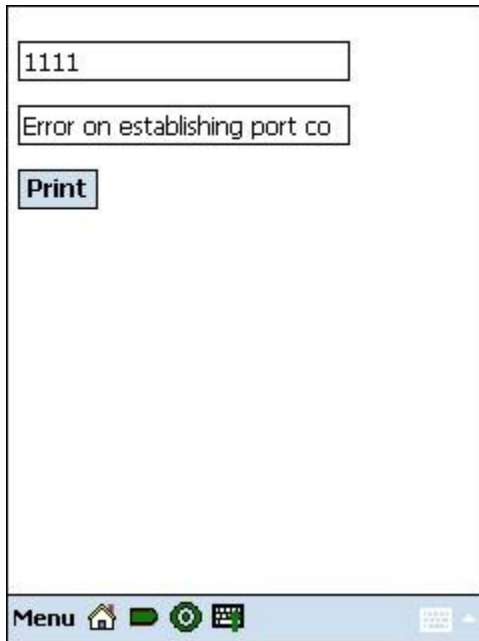
function Imprimer()
{
    PrintObj.PrintFromFile("\\toprint\\text.txt", 1);
}
</SCRIPT>

<BODY>
<OBJECT
    ID="PrintObj" CLASSID="CLSID:5C93AC05-C238-49C1-8CB4-E9D8A9FA242C">
</OBJECT>

<SCRIPT FOR="PrintObj" EVENT="PrintError">
    text1.value = PrintObj.GetLastError();
    text2.value = PrintObj.GetLastErrorDes();
</SCRIPT>

    <P> <INPUT id=text1 name=text1 value=""> </P>
    <P> <INPUT id=text2 name=text2 value=""> </P>
    <input type="button"
        onclick="Imprimer()" value="Imprimer" id=button1 name=button1>
</BODY>
</HTML>

```



Exemple complet

L'exemple suivant montre le fonctionnement d'ActiveX.

```

<HTML>

<HEAD>
    <TITLE> SofToGo.Print </TITLE>
</HEAD>

<SCRIPT ID=clientEventHandlersJS LANGUAGE=javascript>
    function AddData()
    {
        PrintObj.AddData(txtData.value);
        txtRes.value = "AddData OK"
    }

```

```

}
function AddLine()
{
    PrintObj.AddLine(txtLine.value);
    txtRes.value = "AddLine OK"
}
function Print()
{
    PrintObj.Print(txtPCop.value);
    txtRes.value = "Print OK"
}
function Clear()
{
    PrintObj.Clear();
    txtRes.value = "Clear OK"
}
function GetError()
{
    txtErr.value = PrintObj.GetLastError();
    txtErrD.value = PrintObj.GetLastErrorDes();
}
function PrintFromFile()
{
    PrintObj.PrintFromFile(txtPath.value, txtFCop.value);
}
</SCRIPT>

<BODY>
    <OBJECT
        ID="PrintObj"
        CLASSID="CLSID:5C93AC05-C238-49C1-8CB4-E9D8A9FA242C">
    </OBJECT>

    <SCRIPT FOR="PrintObj" EVENT="PrintError">
        GetError()
    </SCRIPT>

    <P>
    Data: <INPUT id=txtData name=txtData size=10 value="Here the text">
    <INPUT type="button" onclick="AddData()" value="AddData" >
    </P>

    <P>
    Line: <INPUT id=txtLine name=txtLine size=10 value="Here the text">
    <INPUT type="button" onclick="AddLine()" value="AddLine" >
    </P>

    <P>
    Copies: <INPUT id=txtPCop name=txtPCop size=2 value="1">
    <INPUT type="button" onclick="Print()" value="Print">
    <INPUT type="button" onclick="Clear()" value="Clear" >
    </P>

    <P>
    <INPUT id=txtRes name=txtRes value="">
    </P>

    <HR>

    <P>
    Path: <INPUT id=txtPath name=txtPath value="\print\test.txt">
    </P>

```

```
</P>  
Copies: <INPUT id=txtFCop name=txtFCop size=2 value="1">  
<INPUT type="button" onclick="PrintFromFile()" value="File Print">  
</P>  
  
<HR>  
  
<P>  
Err:<INPUT id=txtErr name=txtErr size=4 value="">  
<INPUT type="button" onclick="GetError()" value="Get">  
<INPUT id=txtErrD name=txtErrD value="">  
<P>  
  
</BODY>  
  
</HTML>
```

