

Dossier de développement

Emulateur douchette code barre - Simulateur tags

30/06/2015



Application réalisée par .

tel : .
fax : .

Partie 1

En-tête

ECOLOGISTICS

ECO_SCAN

Codes sources

EMULATEUR DOUCHETTE CODE BARRE
SIMULATEUR DE TAGS

Partie 2

Projet

Projet

Code

Initialisation de ECO_SCAN

GLOBAL

```

    gbModeProxy est un booléen = False

HLit (Param,1)
gbModeProxy = Param.Proxy_Int
HCréationsSiInexistant(scénario1)

gcnxCon1 est une Connexion
//if gbModeProxy then
// Proxy(param.Adr_Proxy,param.Port_Proxy,param.Log_Proxy,param.PSW_Proxy)
//END

Connexion_Serveur()
    
```

Projet

Erreurs, warnings et informations

	Détail de l'erreur	Localisation	Ligne	Colon
	Pour optimiser la borne maximale d'une boucle POUR, utilisez la syntaxe optimisée POUR <compteur> = <Min>_A_ <Max> ou bien une variable locale de type entier ou réel.	FEN_Tag_Simu.BTN_SansNom1 Clic sur FEN_Tag_Simu.BTN_SansNom1	10	36
	Pour optimiser la borne maximale d'une boucle POUR, utilisez la syntaxe optimisée POUR <compteur> = <Min>_A_ <Max> ou bien une variable locale de type entier ou réel.	FEN_Tag_Simu.BTN_SansNom1 Clic sur FEN_Tag_Simu.BTN_SansNom1	17	36
	Pour optimiser la borne maximale d'une boucle POUR, utilisez la syntaxe optimisée POUR <compteur> = <Min>_A_ <Max> ou bien une variable locale de type entier ou réel.	FEN_Tag_Simu.BTN_SansNom1 Clic sur FEN_Tag_Simu.BTN_SansNom1	24	36

Projet

Statistiques sur le code

Lignes	%	Lig./tr	
192	44	32	COL_ProcéduresGlobales
59	0	19	FEN_Aquisition1
27	19	9	FEN_Config1
5	0	1	FEN_Mainmenu1
70	0	23	FEN_Tag_Simu
4	0	2	FEN_test_check_digit
13	23	13	ECO_SCAN
364	26	16	

Lignes: Nombre total de lignes de code.

% comm.: Pourcentage de commentaires dans le code.

Lig./trait.: Nombre de lignes de code par traitement.

Partie 3

Fenêtre WinDev

FEN_Mainmenu1

Code

Déclarations globales de FEN_Mainmenu1

```
PROCEDURE MaFenêtre()
```

Fin d'initialisation de FEN_Mainmenu1

```
HLit(Param,1)
```


FEN_Mainmenu1

Code des champs

Clic sur BTN_SansNom1

Ouvre(FEN_Aquisition1)

Clic sur BTN_SansNom2

Ouvre(FEN_Config1)

Clic sur BTN_SansNom3

Ouvre(FEN_Tag_Simu)

FEN_Aquisition1

Code

Déclarations globales de FEN_Aquisition1

PROCEDURE MaFenêtre()

FEN_Aquisition1

Code des champs

Clic sur BTN_SansNom1

```
sligne est une chaîne

IF table_ean..Visible = True THEN
  TABLE_ean..Visible = False
  BTN_Send..Visible = False
  RepriseSaisie(SAI_CodeBarre)
ELSE
  TABLE_ean..Visible=True
  BTN_Send..Visible = True
  table_ean..Multisélection=True
  TableSupprimeTout(TABLE_ean)
  nfc est un entier = fOuvre("ean128.csv", foLectureEcriture)
  IF nfc > 0 THEN
    sligne = fLitLigne(nfc)
    TANTQUE sligne <> EOT
      TableAjoute(TABLE_ean,sligne)
      sligne = fLitLigne(nfc)
  FIN
END
END
```

Clic sur BTN_Send

```
i est un entier
NbSelectionne est un entier
NbSelectionne = TableSelectOccurrence(TABLE_ean)
POUR i = 1 A NbSelectionne
  SAI_CodeBarre = TABLE_ean [TableSelect(TABLE_ean, i)]
  Temporisation(25)
  Exécute("SAI_CodeBarre..sor")
FIN
```

Sortie de SAI_CodeBarre

```
nValret est un entier sur 4 bytes = Val(SAI_CodeBarre)
IF SansEspace(MoiMême) <> "" THEN
  IF nValret <> 0 THEN
    MoiMême+=","
    HRAZ(ReceivePDA)
    ReceivePDA.Data_In=SAI_CodeBarre
    ReceivePDA.TimùeStamp=Today()+Now()
    ReceivePDA.Mode=1
    ReceivePDA.Traite=0
    ReceivePDA.idEquipement = Param.NR_READPOINT
    ReceivePDA.IdScénario = Param.NR_SCENARIO
    HAjoute(ReceivePDA)
  ELSE
    Bip();
    Temporisation (15)
    Bip();
    Temporisation (15)
    Bip();
    Temporisation (15)
    DélaiAvantFermeture(100)
    Info ("press Shift lock on your keyboard"+CR+"Or your scan is not a EAN 128")
    DélaiAvantFermeture()
  END
  SAI_CodeBarre=""
  RepriseSaisie(SAI_CodeBarre)
END
```


FEN_Config1

Code

Déclarations globales de FEN_Config1

```
PROCEDURE MaFenêtre()
```

Fin d'initialisation de FEN_Config1

```
Hlit(Param,1)
IF Param.PSW_Proxy = 1 THEN
  GR_Proxy..Visible = True
ELSE
  gr_proxy..Visible = False
END
FichierVersEcran()
```

FEN_Config1

Code des champs

Clic sur BTN_Save

```
EcranVersFichier  
HModifie(Param)  
HLit (Param,1)  
Connexion_Serveur()  
Ferme
```

A chaque modification de INT_Proxy

```
//if moimeme = 1 then  
// moimeme = 0  
//else  
// moimeme = 1  
//END  
  
IF Moimême = 1 THEN  
GR_Proxy.Visible = True  
gbModeProxy=True  
ELSE  
GR_Proxy.Visible = False  
gbModeProxy=False  
END  
EcranVersFichier  
HModifie(Param)
```


FEN_Tag_Simu

Code

Déclarations globales de FEN_Tag_Simu

PROCEDURE MaFenêtre()

FEN_Tag_Simu

Code des champs

Clic sur BTN_SansNom1

```
nFic est une entier = fOuvre("ean128.csv", foCréation+foLectureEcriture)
ListeSupprimeTout(LISTE_128)

sLigne est une chaîne
CtrlKey est une chaîne = "6"

IF nFic > 0 THEN
  SELON COMBO_GS1
    CAS 1
      FOR i = 1 TO SAI_Quantity
        sLigne = "01"+Gauche(SAI_GTIN,12)+"21"+NumériqueVersChaîne(SAI_Serial+i)

        fEcritLigne(nFic,sLigne)
        ListeAjoute(LISTE_128,sLigne)
      END
    CAS 2
      FOR i = 1 TO SAI_Quantity
        CtrlKey = SSCC_Check_Digit(SAI_GTIN,SAI_Prefix,NumériqueVersChaîne(SAI_Serial+i))
        sLigne = "00"+SAI_GTIN+SAI_Prefix+CtrlKey
        fEcritLigne(nFic,sLigne)
        ListeAjoute(LISTE_128,sLigne)
      END
    CAS 3
      FOR i = 1 TO SAI_Quantity
        CtrlKey=GRAI_Check_Digit(SAI_Prefix,SAI_GTIN)
        sLigne = "80030"+SAI_Prefix+SAI_GTIN+CtrlKey+NumériqueVersChaîne(SAI_Serial+i)
        fEcritLigne(nFic,sLigne)
        ListeAjoute(LISTE_128,sLigne)
      END
    AUTRE CAS
  FIN
  fFerme(nfic)
END
```

Sélection d'une ligne de COMBO_GS1

```
SELON MoiMême
  CAS 1
    SAI_GTIN..Visible = Vrai
    SAI_GTIN..Libellé = "GTIN"
    SAI_Prefix..Visible = Faux
  CAS 3
    SAI_GTIN..Visible = Vrai
    sai_gtin..Libellé = "Asset type"
    SAI_Prefix..Visible = Vrai
  CAS 2
    SAI_GTIN..Visible = Vrai
    SAI_GTIN..Libellé = "Extension"
    SAI_Prefix..Visible = Vrai

  AUTRE CAS
    SAI_GTIN..Visible = Faux
FIN
```

Sortie de SAI_GTIN

```
SELON COMBO_GS1
  CAS 1
    TANTQUE Taille(MoiMême) < 7
      MoiMême += "0"
    FIN
  CAS 3
    TANTQUE Taille(MoiMême) < 5
      MoiMême += "0"
    FIN
    IF Taille (MoiMême) > 5 THEN
      Info("Max 99999")
      MoiMême=""
      RepriseSaisie (SAI_GTIN)
    END
  CAS 2
    MoiMême = Gauche(MoiMême,1)
  AUTRE CAS

FIN
```


FEN_test_check_digit

Code

Déclarations globales de FEN_test_check_digit

PROCEDURE MaFenêtre()

FEN_test_check_digit

Code des champs

Clic sur BTN_SansNom1

```
Info (SSCC_Check_Digit("8","0614141","002244555"))  
Info (SSCC_Check_Digit("0","0718908","562723189"))
```

Partie 4

Collection de procédures

COL_ProcéduresGlobales

Code

Procédure globale Connexion_Serveur

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//Connexion_Serveur ()
//
// Paramètres :
// Aucun
// Valeur de retour :
// Aucune
//
// Exemple :
// Indiquez ici un exemple d'utilisation.
//
PROCEDURE Connexion_Serveur()

HLit(Param,1)
HDécritConnexion(gcnxCon1,"Admin","",Param.IP_ALE,"sv_ecolog",hAccèsHFClientServeur)
//
//
IF HouvreConnexion(gcnxCon1) THEN

    HChangeConnexion(Receive1,gcnxCon1)
    HChangeConnexion(equipements,gcnxCon1)
    HCréationSiInexistant(Receive1)

END
```

Procédure globale EPCIS__Sending

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//EPCIS__Sending ()
//
// Paramètres :
// Aucun
// Valeur de retour :
// Aucune
//
// Exemple :
```

```
// Indiquez ici un exemple d'utilisation.
///// Procédure automatique :
// La procédure est exécutée automatiquement, après le code d'initialisation du projet, avec un différé de 5 secondes
// Elle s'exécutera dans un thread (sans avoir besoin d'appeler la fonction ThreadExecute), sans utilisation de HFSQL
// Elle sera répétée en boucle, en attendant 3 secondes entre chaque appel
//
```

```
PROCEDURE EPCIS__Sending()
```

```
i est un entier
ip_range est une chaîne = ""
sLib_ip est une chaîne = ""
sLib_ip = NetAdresseIP()
```

```
FOR i = 1 TO 3
    ip_range += ExtraitChaîne(sLib_ip,i,",")+","
END
```

```
/////if position(param.IP_ALE,ip_range) > 0 then
// if HDécritConnexion(cnxEpcis_,"admin","",param.IP_ALE,"sv_ecolog",hAccèsHFClientServeur) THEN
//     if HChangeConnexion(Receive1,cnxEpcis_) then
//         IF HlitPremier(ReceivePDA,IDReceive) THEN
//             TANTQUE PAS HEnDehors(ReceivePDA)
//                 HRAZ(Receive1)
//                 Receive1.Data_In = ReceivePDA.Data_In
//                 Receive1.idEquipement = ReceivePDA.idEquipement
//                 Receive1.IdScénario = ReceivePDA.IdScénario
//                 Receive1.IP_Adresse = ReceivePDA.IP_Adresse
//                 Receive1.Mode = ReceivePDA.Mode
//                 Receive1.TimùeStamp = ReceivePDA.TimùeStamp
//                 Receive1.Traite = ReceivePDA.Traite
//                 IF HAjoute(Receive1) THEN
//                     HSupprime(ReceivePDA)
//                 END
//                 HlitSuivant(ReceivePDA,IDReceive)
//             END
//         END
//     end
// END
/////end
```

Procédure globale GRAI_Check_Digit

```

PROCEDURE GRAI_Check_Digit(sai_pc,sai_asset)
svalRet est une chaîne="0"

d1,d2,d3,d4,d5,d6,d7,d8,d9,d10,d11,d12,d13 sont des entiers
TANTQUE Taille (SAI_PC) < 7
  SAI_PC = "0"+SAI_PC
END
TANTQUE Taille (SAI_ASSET) < 5
  SAI_ASSET = "0"+SAI_ASSET
END

d1 = Val(Milieu(SAI_PC,1,1))
d2 = Val(Milieu(SAI_PC,2,1))
d3 = Val(Milieu(SAI_PC,3,1))
d4 = Val(Milieu(SAI_PC,4,1))
d5 = Val(Milieu(SAI_PC,5,1))
d6 = Val(Milieu(SAI_PC,6,1))
d7 = Val(Milieu(SAI_PC,7,1))
d8 = Val(Milieu(SAI_ASSET,1,1))
d9 = Val(Milieu(SAI_ASSET,2,1))
d10 = Val(Milieu(SAI_ASSET,3,1))
d11 = Val(Milieu(SAI_ASSET,4,1))
d12 = Val(Milieu(SAI_ASSET,5,1))

a1 est un entier = 3*(D2 + D4 + D6 + D8 + D10 + D12)
a2 est un entier = D1+ D3 + D5 + D7 + D9 + D11

d13 = modulo((10 - (a1 + a2) ),10)
IF d13 < 0 THEN
  d13+=10
END
svalret = NumériqueVersChaîne(d13, "1d")
RENOYER(svalret)

```

Procédure globale SSCC_Check_Digit

```

PROCEDURE SSCC_Check_Digit(Sai_Ext,sai_pc,sai_serial)
svalRet est une chaîne="0"
sscc est une chaîne
i est un entier
d est un tableau de 18 entier

```

```

t est un entier = 17 - (Taille(sai_pc+1))
IF Sai_Ext = "" THEN Sai_Ext = "0"
sai_ext = Gauche(Sai_Ext,1)
TANTQUE Taille (sai_serial) < t
    sai_serial = "0"+sai_serial
FIN
ssccc = sai_ext+sai_pc+sai_serial
FOR i = 1 TO 17
    d[i] = Val(Milieu(ssccc,i,1))
END

a2 est un entier = (D[2] + D[4] + d[6] + D[8] + D[10]+ D[12]+ D[14]+ D[16])
a1 est un entier = 3*(d[1] + d[3] + d[5] + d[7]+ d[9]+ d[11]+ d[13]+ d[15]+ d[17])

d[18] = modulo((10 - (a1 + a2) ),10)
IF d[18] < 0 THEN
    d[18]+=10
END
svalret = sai_serial+NumériqueVersChaîne(d[18], "1d")
REVOYER(svalret)

```

Procédure globale XML__Compute

```

PROCEDURE XML__Compute()
//sLigne est une chaîne
//
//nFic est un entier = fouvre(gsPathFileCapture,foCréation+foLectureEcriture)
////nFic est un entier = fouvre("c:\data\epcis.txt",foCréation+foLectureEcriture)
//if nFic > 0 then
//    sligne = "<?xml version="+caract(34)+"1.0"+caract(34)+ " encoding="+caract(34)+"UTF-8"+caract(34)+"?>"; fEcritLigne(nFic,sligne)
//    sligne = "<epcis:EPCISDocument xmlns:epcis="+Caract(34)+"urn:epcglobal:epcis:xsd:1"+Caract(34)+
//
//        " xmlns:xsi="+caract(34)+"http://www.w3.org/2001/XMLSchema-instance"+caract(34)+"
//        creationDate="+caract(34)+DateVersChaîne(today, "AAAA-MM-JJ")+ "T"+
//
//        heureverschaîne(now, "HH:MM:SS")+ ".016+01:00"+caract(34)+ " schemaVersion="+caract(34)+"1.0"+caract(34)+ " xmlns:myNs="+caract(34)+
//        "http://my.unique.namespace"+caract(34)+">"; fEcritLigne(nFic,sligne)
//        // ="2014-11-26T10:42:15.016+01:00" schemaVersion="1.0" xmlns:myNs="http://my.unique.namespace">
//    sligne = "<EPCISBody>"; fEcritLigne(nFic,sligne)
//    sLigne = "<EventList>"; fEcritLigne(nFic,sLigne)
//    sLigne = "<ObjectEvent>"; fEcritLigne(nFic,sLigne)
//
//

```

```

sLigne = "<eventTime>" + DateVersChaîne(Today, "AAAA-MM-JJ") + "T" + HeureVersChaîne(Now, "HH:MM:SS") + "Z</eventTime>" ; fEcritLigne(nFic, sLigne)
// sLigne = "<eventTimeZoneOffset>+00:00</eventTimeZoneOffset>" ; fEcritLigne(nFic, sLigne)
// sLigne = "<epcList>" ; fEcritLigne(nFic, sLigne)
// sLigne = "<epc>urn:epc:id:sgtin:3270201.080021.91200005</epc>"; fEcritLigne(nFic, sLigne)
// sLigne = "</epcList>" ; fEcritLigne(nFic, sLigne)
// sLigne = "<action>ADD</action>"; fEcritLigne(nFic, sLigne)
// sLigne = "<bizStep>urn:epcglobal:fmcg:bizstep:receiving</bizStep>"; fEcritLigne(nFic, sLigne)
// sLigne = "<disposition>urn:fosstrak:demo:disp:fmcg:pendingQA</disposition>"; fEcritLigne(nFic, sLigne)
// sLigne = "<readPoint>"; fEcritLigne(nFic, sLigne)
// sLigne = "<id>urn:fosstrak:demo:fmcg:ssl:0037000.00729.210,432</id>"; fEcritLigne(nFic, sLigne)
// sLigne = "</readPoint>"; fEcritLigne(nFic, sLigne)
// sLigne = "<bizLocation>"; fEcritLigne(nFic, sLigne)
// sLigne = "<id>urn:fosstrak:demo:fmcg:ssl:0037000.00729.210</id>"; fEcritLigne(nFic, sLigne)
// sLigne = "</bizLocation>"; fEcritLigne(nFic, sLigne)
// sLigne = "</ObjectEvent>"; fEcritLigne(nFic, sLigne)
// sLigne = "</EventList>"; fEcritLigne(nFic, sLigne)
// sLigne = "</EPCISBody>"; fEcritLigne(nFic, sLigne)
// sLigne = "</epcis:EPCISDocument>"; fEcritLigne(nFic, sLigne)
// //sLigne = ""
// fferme(nfic)
// xml__post(gsPathEPCIS_Server, gsPathFileCapture)
//END

```

Procédure globale XML__POST

PROCEDURE XML__POST(UrL, FileXML)

```

tosend est un Buffer
url=""
i,j est un entier
a1 est une chaîne
sLigne est une chaîne
nFic est une entier =ouvre(FileXML, foLectureEcriture)
IF nFic > 0 THEN
    sLigne = fLitLigne(nFic)
    TANTQUE sLigne <> EOT

        FOR i = 1 TO Taille (sLigne)
            j++
            a1 = Asc(Milieu(sLigne, i, 1))
            tosend [[j]] = Val(a1)//NumériqueVersChaîne(asc(Milieu(sLigne, i, 1)), "02X")
        FIN
    sLigne = fLitLigne(nFic)
END
END

```

```
//hraz(msg_Entrant)
//msg_Entrant.Contenu=tosend
//hajoute(msg_Entrant)

//DélaiAvantFermeture(150)
//HTTPCréeFormulaire("Form")
//HTTPAjouteParamètre("Form","",tosend)
//IF HTTPEnvoieFormulaire("Form",Url,httpPost, "", "", "text/xml") THEN
//  Info ("EPCIS OK")
//ELSE
//  Info("EPICIS CAPTURE FAILED")
//END
//DélaiAvantFermeture()
```

Partie 5

Table des matières

Table des matières

Projet ECO_SCAN

3 En-tête Partie 1

3 ○ **En-tête**

5 Projet Partie 2

5 ○ **Code**

5 ○ **Erreurs, warnings et informations**

6 ○ **Statistiques sur le code**

8 Fenêtre WinDev Partie 3

8 ○ **FEN_Mainmenu1**

8 ○ Code

9 ○ Code des champs

10 ○ **FEN_Aquisition1**

11 ○ Code

12 ○ Code des champs

14 ○ **FEN_Config1**

15 ○ Code

16 ○ Code des champs

17 ○ **FEN_Tag_Simu**

18 ○ Code

19 ○ Code des champs

21 ○ **FEN_test_check_digit**

22 ○ Code

23 ○ Code des champs

25 Collection de procédures Partie 4

25 ○ **COL_ProcéduresGlobales**

25 ○ Code