

# Development file



## Barcode scanner emulator Tags simulator

---

30/06/2015



Application created by .

tel : . fax : .

## Part 1

### Heading

Part 1 Heading

ECOLOGISTICS

ECO\_SCAN

## Sources codes

BARCODE SCANNER EMULATOR TAGS SIMULATOR

## Part 2

### Project

### Project

Code

Initialisation of ECO\_SCAN

```
GLOBAL gbModeProxy est un booléen =  
    False
```

## Project

## Statistics on the code

```
HLit (Param,1)
gbModeProxy = Param.Proxy_Int
HCréationSiInexistant(scénario1)

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

Connexion_Serveur()
```

## Project

## Errors, warnings and informations

Detail of errors			Localisation	Line
Lines	%	Lig./tr		Colon
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	
			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
			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
			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
4	0	2	FEN_test_check_digit	
13	23	13	ECO_SCAN	
<b>364</b>	<b>26</b>	<b>16</b>		

**Lignes:** Total number of the code lines  
**Lig./trait.:** Number of lines per processing.

**% comm.:** % of comment in the code.





## Part 3

### WinDev window

Part 3 WinDev window

### FEN\_Mainmenu1

Code

Global declarations of FEN\_Mainmenu1

```
PROCEDURE MaFenêtre()
```



HLit(Param,1)

## FEN\_Mainmenu1

Code of fields

Click on BTN\_SansNom1

`Ouvre(FEN_Aquisition1)`

Click on BTN\_SansNom2

`Ouvre(FEN_Config1)`

Click on BTN\_SansNom3

`Ouvre(FEN_Tag_Simu)`



## FEN\_Aquisition1

Code

### Global declarations of FEN\_Aquisition1

```
PROCEDURE MaFenêtre()
```

## FEN\_Aquisition1

### Code of fields

#### Click on 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)
  nfic est un entier = fOuvre("ean128.csv",foLectureEcriture)
  IF nfic>0 THEN sligne =
    fLitLigne(nfic)
    TANTQUE sligne<>EOT
      TableAjoute(TABLE_ean,sligne)
      sligne = fLitLigne(nfic)
  FIN
END
END
```

#### Click on 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)
```

▼ Project ECO\_SCAN

```
Exécute("SAI_CodeBarre..sor")
```

```
FIN
```

Exit of SAI\_CodeBarre

Part 3

Code of fields

```
nvalret est un entiersur4bytes = 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
```

Part 3

WinDev window

## FEN\_Config1

Code

### Global Declarations of FEN\_Config1

```
PROCEDURE MaFenêtre()
```

### End of initialisation of 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 of fields

Click on BTN\_Save

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

In each modification of INT\_Proxy

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

Part 3 WinDev window



## FEN\_Tag\_Simu

Code

### Global Declarations of FEN\_Tag\_Simu

```
PROCEDURE MaFenêtre()
```

## FEN\_Tag\_Simu

Code of fields

Click on 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
```

Selection of a line of 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
```

---

**Exit of 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

### Global Declarations of FEN\_test\_check\_digit

```
PROCEDURE MaFenêtre()
```

## FEN\_test\_check\_digit

Code of fields

Click on BTN\_SansNom1

Info (SSCC\_Check\_Digit("8","0614141","002244555"))

Info (SSCC\_Check\_Digit("0","0718908","562723189"))

## Part 4

# Collection of procedures

## COL\_ProcéduresGlobales

Code

### Global Procedure 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
```

---

### Global Procedure 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.
n. ////
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
```

### Global Procedure 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) <7SAI_PC =
  "0"+SAI_PC
END
```

```
TANTQUE Taille (SAI_ASSET)
<5SAI_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")
REVOYER(svalret)

```

---

#### Global Procedure 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 18entier 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)
<tsai_serial = "0"+sai_serial FIN
SSCC =
sai_ext+sai_pc+sai_serial

FOR i = 1 TO 17 d[i] =
Val(Milieu(ssscc,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)

```

### Global Procedure 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:fmcg: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
```

---

**Global Procedure XML\_\_POST**

---

```
PROCEDURE XML__POST(Url,FileXML)
tosend est un
Bufferurl="" i,j est
un entier a est une
chaîne sLigne est
une chaîne
nFic est une entier = fOuvre(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
//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()
```

# Table of contents

# Part 5

# Table of contents

## Project ECO\_SCAN

▼ Project ECO\_SCAN

3 Heading

Part 1

3 ○ **Heading**

5 Project

Part 2

5 ○ **Code**

5 ○ **Errors, warnings and informations**

6 ○ **Statistics on the code**

8 WinDev Window

Part 3

8 ○ **FEN\_Mainmenu1**

8 ○ Code

9 ○ Code of fields

10 ○ **FEN\_Aquisition1**

11 ○ Code

12 ○ Code of fields

14 ○ **FEN\_Config1**

15 ○ Code

16 ○ Code of fields

17 ○ **FEN\_Tag\_Simu**

18 ○ Code

19 ○ Code of fields

21 ○ **FEN\_test\_check\_digit**

22 ○ Code

23 ○ Code of fields

25 Collection de procédures

Part 4

25 ○ **COL\_ProcéduresGlobales**

25 ○ Code