

Development file



30/06/2015

Application created by (null)

tel : (null) fax : (null)

Part 1

Heading

Eco_Scan_PDA codes sources file

Part 1 Heading

Part 2

Project

Project

Code

Initialisation of Ecologistics

GLOBAL

```
// HCréationSiInexistent(msg_Entrant)
```

30/06/2015

Project

Statistics on the code

```

gsbizstepname est une chaîne
gsScenarioName est une chaîne
HLit (Param,1)
HCréationSiInexistant(scénario1)

gcnxCon1 est une connexion

HDécritConnexion(gcnxCon1,"Admin","",Param.IP_ALE,"Sv_Ecolog",hAccèsHFClientServeur)
//
//
IF HOuvreConnexion(gcnxCon1) THEN
    HChangeConnexion(Receive1,gcnxCon1)
    HCréationSiInexistant(Receive1)
END
    
```

Project

Errors, warnings and informations

Details of error		Localisation	Line	Colon
Lines	%			

	La hauteur du champ 'FEN_Config.LIB_SansNom1' est trop petite pour afficher correctement le libellé en Français	FEN_Config.LIB_SansNom1 (null)	1	1
--	---	-----------------------------------	---	---

137	61	34	COL_ProcéduresGlobales
17	0	4	FEN_Aquisition
17	53	5	FEN_Config
26	38	6	FEN_Mainmenu
19	16	19	Ecologistics
212	50	13	

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

% comm.: % of comment in the code.

Part 3

WinDev Mobile Window

Part 3 WinDev Mobile Window

FEN_Mainmenu

Code

Global declarations of FEN_Mainmenu

```
PROCEDURE MaFenêtre()
```

Fin d'initialisation de FEN_Mainmenu

```
//HCréation(ReceivePDA)
HCréationSiInexistant(Param)
IF HNbEnr(Param) = 0 THEN
  HRAZ(Param)
  Param.IP_ALE = "192.168.14.100"
  Param.NR_SCENARIO = 1
  Param.NR_READPOINT = 1
  HAjoute(Param)
  Ouvre(FEN_Config)
ELSE
  HLit(Param, 1)
END
//scle est une chaîne = Key_Eqt__Compute(param.NR_SCENARIO,param.NR_READPOINT)
//HLitRecherche(equipements,IP_Address,scle)
//if htrouve then
// gsbizstepname = ExtraitChaîne(equipements.BizStepName,1,":",DepuisFin)
// HLitRecherche(scénario1,idScénario,param.NR_SCENARIO)
// if htrouve then
//   gsScenarioName = scénario1.Name
// end
//end
```

Part 3 Code of fields

FEN_Mainmenu

Code of fields

Clic sur BTN_SansNom1

```
Ouvre(FEN_Aquisition)
```

Clic sur BTN_SansNom2

```
Ouvre(FEN_Config)
```


FEN_Aquisition

Code

Global déclarations of FEN_Aquisition

```
PROCEDURE MaFenêtre()
```

End of initialisation of FEN_Aquisition

```
RepriseSaisie(SAI_CodeBarre)
```

FEN_Aquisition

Code of fields

Initialisation of LIB_Descr

```
MoiMême = gsScenarioName+CR+gsbizstepname+CR+"Readpoint:" + Param.NR_READPOINT
```

Sortie de SAI_CodeBarre

```
IF SansEspace(MoiMême) <> "" THEN
  MoiMême += ";"
  HRAZ(ReceivePDA)
  ReceivePDA.Data_In=SAI_CodeBarre
  ReceivePDA.TimeStamp=Today()+Now()
  ReceivePDA.Mode=1
  ReceivePDA.Traite=0
  ReceivePDA.idEquipement = Param.NR_READPOINT
  ReceivePDA.IdScénario = Param.NR_SCENARIO
  HAjoute(ReceivePDA)
  SAI_CodeBarre=""
  RepriseSaisie(SAI_CodeBarre)
END
```

FEN_Config

Code

Global declarations of FEN_Config

```
PROCEDURE MaFenêtre()
```

End of initialisation of FEN_Config

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

FEN_Config

Code of fields

Clic sur BTN_SansNom1

```
EcranVersFichier
HModifie(Param)
HLit(Param,1)
//scle est une chaine = Key_Eqt__Compute(param.NR_SCENARIO,param.NR_READPOINT)
//HLitRecherche(equipements,IP_Address,scle)
//IF HTrouve THEN
// gsbizstepname = ExtraitChaîne(equipements.BizStepName,1,DepuisFin)
// HLitRecherche(scenario1,idScenario,Param.NR_SCENARIO)
// IF HTrouve THEN
//     gsScenarioName = scenario1.Name
// END
//END
Ferme
```

Part 4

Collection of procedures

COL_ProcéduresGlobales

Code

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.
//// 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()
```

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

```
FOR i = 1 TO 2 ip_range +=
  ExtraitChaîne(sLib_ip,i, ".")+ "."
END
```

```
//if position(param.IP_ALE,ip_range) > 0 then
// info("range")
```

```
IF HDécritConnexion(cnxEpcis_,"Admin","",Param.IP_ALE,"Sv_Ecolog",hAccèsHFClientServeur) THEN IF
  HOuvreConnexion(cnxEpcis_) THEN
    //info ("ouvre")
    IF HChangeConnexion(Receive1,cnxEpcis_) THEN
      IF HlitPremier(ReceivePDA,IDReceive) THEN
        TANTQUE PAS HEnDehors(ReceivePDA)
          Receive1.Data_In = ReceivePDA.Data_In
          //Info ("dans process")
          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
// end
///end
```

Global Procedure Key_Eqt_Compute

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//[ <Résultat> = ] Key_Eqt_Compute (<sc>, <eq>)
//          //
Paramètres :
// sc : <indiquez ici le rôle de sc>
// eq : <indiquez ici le rôle de eq> //
valeur de retour :
// chaîne ANSI : // Aucune
//          //
Exemple :
// Indiquez ici un exemple d'utilisation.
//
PROCEDURE Key_Eqt_Compute(sc,eq)
```

sva1ret est une chaîne = NumériqueVersChaîne(sc,"04d")+NumériqueVersChaîne(eq,"04d")

REVOYER(sva1ret)

Global procedure XML__Compute

30/06/2015

18

```
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
```

Global procedure XML__POST

```
PROCEDURE XML__POST(Url,FileXML)
```

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

    FORi = 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","",msg_Entrant.Contenu)
//IF HTTPEnvoieFormulaire("Form",Url,httpPost, "", "", "text/xml") THEN
// Info ("EPCIS OK")
//ELSE
// Info("EPICIS CAPTURE FAILED")
//END
//DélaiAvantFermeture()
```


Part 5

Table of contents

Table of contents

Project Ecologistics

▼ Project Ecologistics

3 Heading

Part 1

3 ○ Heading

5 Project

Part 2

5 ○ Code

5 ○ Errors, warnings and information

6 ○ Statistics on the code

8 WinDev Mobile Window

Part 3

8 ○ FEN_Mainmenu

8 ○ Code

9 ○ Code of fields

10 ○ FEN_Aquisition

11 ○ Code

12 ○ Code of fields

13 ○ FEN_Config

14 ○ Code

15 ○ Code of fields

17 Collection of procedures

Part 4

17 ○ COL_ProcéduresGlobales

17 ○ Code