Dossier de développement

Système d'acquisition de données des équipements

30/06/2015

Ecologistics

Application réalisée par .

tel : .
fax : .

Partie 1

En-tête

ECOLOGISTICS

CODES SOURCES

Service SV_ECOLOG

Service d'acquisition de données
des équipements

Partie 2

Projet

Projet

Code

Initialisation de SV_Ecolog

```
GLOBAL
    gnLastDH est un entier = 0
    gsAlien_Ip est une chaîne
    gsAlien_Login est une chaîne
    gsAlien_Paswword est une chaîne
    gsALien_FileNameResut est une chaîne
    gnAlien_RF_Attenuation est un entier
    gclMyClsReader est une clsReader
    gsMsgRet est une chaîne
    gnReadPoint est un entier =4
    gnDispo est un entier=5
    gcnxSv est une Connexion
    gsPathEPCIS_Server est une chaîne = "http://127.0.0.1:8080/epcis-repository-0.5.0/capture"

    HCréationSiInexistant(Msg_entrant)
    HCréationSiInexistant(Alien_Rfid)

IF HOuvreConnexion (CnxEcolog) THEN
    HChangeConnexion(Msg_entrant,CnxEcolog)

    HSupprimeTout(Alien_Rfid)


END
```

Exécution du service de SV_Ecolog (appelé en boucle)

```
Traitement_Msg()
```

Projet

Statistiques sur le code

Lignes	%	Lig./tr	
615	41	30	COL_ProcéduresGlobales
32	0	16	FEN_test
50	2	25	SV_Ecolog
639	**40**	**26**	

Lignes: Nombre total de lignes de code.
Lig./trait.: Nombre de lignes de code par traitement.
% comm.: Pourcentage de commentaires dans le code.

Partie 3

Analyse

Analyse

Graphe


srv_epcis
idSrv_EPCIS
...
MySQL
eqts_log
idEqts_Log
MySQL
scenario
idScénario
...
MySQL
prefix
idPrefix
Prefix_Code
Prefix_Length
equipements
idEquipments
HFSQL C/S
srv_epcisf
idSrv_EPCIS
...
HFSQL C/S

Analyse

Informations générales

SV_Ecolog.wda	

Répertoire	C:\Mes Projets\SV_Ecolog\SV_Ecolog.ana\
Répertoire des fichiers de données	Répertoire de l'application

N°génération	Nb. fichiers	Nb. rubriques	Nb. liaisons	Nb. connexions	Nb. groupes
25	11	57	0	3	0

Analyse

Dictionnaire des rubriques

Rubrique	Type	Taille	Clé	Clé	Utilisée par...
Action	Chaîne unicode	45			equipements
Autre_Data	Chaîne	100			Receive
	Chaîne	100			ReceivePDA
BiszStepID	Entier sur 4 octets				equipements
BizDipositionID	Entier sur 4 octets				equipements
BizDispositionName	Chaîne unicode	45			equipements
BizLobGLN					<Inutilisée>
BizLocGLN	Chaîne unicode	100			equipements
BizLocID	Chaîne unicode	100			equipements
BizLocName	Chaîne unicode	100			equipements
BizStepName	Chaîne unicode	45			equipements
CleComp1	Clé composée	8		O	equipements
Comment	Image (mémo binaire)				scénario
Contenu	Mémo texte				Msg_entrant

Rubrique	Type	Taille	Clé	Clé	Utilisée par...
Contenut					<Inutilisée>
Data_In	Mémo texte				Receive
	Chaîne	50			ReceivePDA
Data_Out	Mémo texte				Receive
	Chaîne	50			ReceivePDA
DATE	Date et Heure			O	Alien_Rfid
dns	Chaîne unicode	100		O	srv_epcis
	Chaîne unicode	100		O	srv_epcis1
EPCIS_URL_Capture	Chaîne unicode	100			equipements
Equipement_Type	Entier sur 4 octets			O	equipements
GPS	Chaîne	50			Receive
	Chaîne	50			ReceivePDA
IDAlien_Rfid	Identifiant automatique (4 octets)		O		Alien_Rfid
idEPCIS	Entier sur 4 octets			O	eqts_log
	Entier sur 4 octets			O	equipements
idEqts_Log	Identifiant automatique (4 octets)		O		eqts_log
idEquipement	Entier sur 4 octets			O	Receive
	Entier sur 4 octets			O	ReceivePDA
idEquipments	Entier sur 4 octets			O	eqts_log
	Identifiant automatique (4 octets)		O		equipements
IDMsg_entrant	Identifiant automatique (8 octets)		O		Msg_entrant
IDParam	Identifiant automatique (8 octets)		O		Param
idPrefix	Identifiant automatique (4 octets)		O		prefix
IDReceive	Identifiant automatique (8 octets)		O		Receive
	Identifiant automatique (8 octets)		O		ReceivePDA
IDReeceive					<Inutilisée>
IdScénario	Entier sur 4 octets			O	Receive
	Entier sur 4 octets			O	eqts_log
	Identifiant automatique (4 octets)		O		scénario
	Entier sur 4 octets			O	ReceivePDA
	Entier sur 4 octets			O	equipements
idSrv_EPCIS	Identifiant automatique (4 octets)		O		srv_epcis
	Identifiant automatique (4 octets)		O		srv_epcis1
IP_Address	Chaîne unicode	45		O	srv_epcis

Rubrique	Type	Taille	Clé	Clé	Utilisée par...
	Chaîne unicode	45		O	equipements
	Chaîne unicode	45		O	srv_epcis1
IP_Adresse	Chaîne	50		O	Alien_Rfid
	Chaîne	50		O	Receive
	Chaîne	50		O	ReceivePDA
IP_ALE	Chaîne	50			Param
Mode	Entier non signé sur 1 octet				Receive
	Entier non signé sur 1 octet				ReceivePDA
Model	Chaîne unicode	45			equipements
Name	Chaîne unicode	45			scénario
Nr_Antennas	Entier sur 4 octets				equipements
NR_READPOINT	Entier sur 4 octets				Param
NR_SCENARIO	Entier sur 4 octets				Param
NumFabr	Chaîne	50	O		Alien_Rfid
Port_Nr	Entier sur 4 octets				equipements
Prefix_Code	Chaîne unicode	45	O		prefix
Prefix_Length	Entier sur 4 octets				prefix
RaedPointGLN	Chaîne unicode	100			equipements
ReadPointID	Entier sur 4 octets				equipements
ReadPointName	Chaîne unicode	100			equipements
Rfid_Attenuation	Entier sur 4 octets				equipements
TimeStamp	Date et Heure			O	eqts_log
Timing	Entier sur 4 octets				equipements
TimùeStamp	Date et Heure				Receive
	Date et Heure				ReceivePDA
Traite	Booléen			O	Msg_entrant
	Booléen			O	Alien_Rfid
	Booléen			O	Receive
	Booléen			O	ReceivePDA
Triggers mode	Entier sur 4 octets				equipements
URL_EPCIS	Chaîne	50			Receive
	Chaîne	50			ReceivePDA
XML	Image (mémo binaire)				eqts_log

Analyse

Connexions

Connexion	Source de données	Utilisateur	Provider	Tm.	Tm.
CnxEcolog **CnxEcolog**	127.0.0.1:4900	admin	WinDevClientServeurHF	30	30
Utilisée par...	Msg_entrant	Alien_Rfid	Receive		
	equipements	srv_epcis1			
MySQL_Ecolog_workbench **Connexion à '127.0.0.1'**	127.0.0.1	root	WinDevMySQL	30	30
Utilisée par...	eqts_log	scénario	srv_epcis		
Interop **Connexion à '127.0.0.1'**	127.0.0.1	root	WinDevMySQL	30	30

Tm. Conn*: Timeout de connexion **Tm. Exec*:** Timeout d'exécution

Analyse

Fichiers et rubriques

	Abrévi	Id.	N°gén	Taille	+ de	Espac	Réplic	Jnl.	Jnl.	Sécuri	Crypt.	Crypt.	Crypt.	Compr	Type
Alien_Rfid		o	5	124											HFSQL Client/Serveur

	Abrévi	Id.	N°gén	Taille	+ de	Espac	Réplic	Jnl.	Jnl.	Sécuri	Crypt.	Crypt.	Crypt.	Compr	Type
eqts_log		O	1	32											Accès natif / Autre accès OLE DB
equipements		O	8	1732											HFSQL Client/Serveur
Msg_entrant		O	2	26	O						O	O	O		HFSQL Client/Serveur
Param		O	1	76	O						O	O	O		HFSQL Classic
prefix		O	1	109										O	HFSQL Classic
Receive		O	4	306	O						O	O	O		HFSQL Client/Serveur
ReceivePDA		O	4	392	O						O	O	O		HFSQL Classic
scénario		O	1	104											Accès natif / Autre accès OLE DB
srv_epcis		O	2	298											Accès natif / Autre accès OLE DB
srv_epcis1		O	3	307											HFSQL Client/Serveur

Espaces *: Complétion des chaînes par des espaces
Jnl. Lect/Ecr *: Journalisation des lectures et écitures
Jnl. Ecr *: Journalisation des écritures
Sécurité *: Mode sécurité renforcée

Alien_Rfid

Fichiers et rubriques

Informations générales

Alien_Rfid	Alien_Rfid

Nom sur disque	Alien_Rfid.FIC
Connexion	CnxEcolog

Rubriques du fichier Alien_Rfid

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
IDAlien_Rfid	Identifiant de Alien_Rfid	Identifiant automatique (4 octets)		O		■	
NumFabr	Numfabr	Chaîne	50	O		■	
DATE	DATE	Date et Heure (aaaammjjhhmmssccc)			O	■	0000000000000
IP_Adresse	Ip_adresse	Chaîne	50		O	■	
Traite	Traite	Booléen			O	■	0

eqts_log

Fichiers et rubriques

Informations générales

eqts_log	eqts_log (partagé)
Connexion	Connexion à '127.0.0.1'

Rubriques du fichier eqts_log

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
idEqts_Log	idEqts_Log	Identifiant automatique (4 octets)		O		■	
XML	XML	Image (mémo binaire)					
TimeStamp	TimeStamp	Date et Heure (aaaammjjhhmmssccc)			O	■	
idEquipments	idEquipments	Entier sur 4 octets			O	■	
idEPCIS	idEPCIS	Entier sur 4 octets			O	■	
idScenario	idScenario	Entier sur 4 octets			O	■	

equipements

Fichiers et rubriques

Informations générales

equipements	equipements

Nom sur disque	equipements.fic
Connexion	CnxEcolog

Rubriques du fichier equipements

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
idEquipments	idEquipments	Identifiant automatique (4 octets)		O		■	
idScenario	idScenario	Entier sur 4 octets			O	■	
idEPCIS	idEPCIS	Entier sur 4 octets			O	■	
BizLocName	BizLocName	Chaîne unicode	100				
BizLocID	BizLocID	Chaîne unicode	100				
BizLocGLN	BizLocGLN	Chaîne unicode	100				
ReadPointName	ReadPointName	Chaîne unicode	100				
ReadPointID	ReadPointID	Entier sur 4 octets					
RaedPointGLN	RaedPointGLN	Chaîne unicode	100				
Equipement_Type	Equipement_Type	Entier sur 4 octets			O	■	
Model	Model	Chaîne unicode	45				
IP_Address	IP_Address	Chaîne unicode	45		O	■	
Port_Nr	Port_Nr	Entier sur 4 octets					
Nr_Antennas	Nr_Antennas	Entier sur 4 octets					
Triggers mode	Triggers mode	Entier sur 4 octets					
Timing	Timing	Entier sur 4 octets					
Action	Action	Chaîne unicode	45				
BiszStepID	BiszStepID	Entier sur 4 octets					
BizStepName	BizStepName	Chaîne unicode	45				

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
BizDipositionID	BizDipositionID	Entier sur 4 octets					
BizDispositionName	BizDispositionName	Chaîne unicode	45				
EPCIS_URL_Capture	EPCIS_URL_Capture	Chaîne unicode	100				
Rfid_Attenuation	Rfid_Attenuation	Entier sur 4 octets					
CleComp1	CleComp1	Clé composée : idScenario+ReadPointID	8		○	■	

Msg_entrant

Fichiers et rubriques

Informations générales

Msg_entrant	Msg_entrant

Nom sur disque	Msg_entrant.fic
Connexion	CnxEcolog

Rubriques du fichier Msg_entrant

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
IDMsg_entrant	Identifiant de Msg_entrant	Identifiant automatique (8 octets)		O		■	
Contenu	Contenu	Mémo texte					
Traite	Traite	Booléen			O	■	0

Param

Fichiers et rubriques

Informations générales

Param	Param
Nom sur disque	Param.fic

Rubriques du fichier Param

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
IDParam	Identifiant de Param	Identifiant automatique (8 octets)		O		■	
IP_ALE	Ip_ale	Chaîne	50				
NR_SCENARIO	Nr_scenario	Entier sur 4 octets					0
NR_READPOINT	Nr_readpoint	Entier sur 4 octets					0

prefix

Fichiers et rubriques

Informations générales

prefix	prefix (importé)
Nom sur disque	prefix.FIC

Rubriques du fichier prefix

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
idPrefix	idPrefix	Identifiant automatique (4 octets)		O		■	
Prefix_Code	Prefix_Code	Chaîne unicode	45	O		■	
Prefix_Length	Prefix_Length	Entier sur 4 octets					

Receive

Fichiers et rubriques

Informations générales

Receive	Receive

Nom sur disque	Reeceive.fic
Connexion	CnxEcolog

Rubriques du fichier Receive

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
IDReceive	Identifiant de Receive	Identifiant automatique (8 octets)		O		■	
Mode	Mode	Entier non signé sur 1 octet					0
Data_In	Data_in	Mémo texte					
Data_Out	Data_out	Mémo texte					
TimùeStamp	Timùestamp	Date et Heure (aaaammjjhhmmssccc)					0000000000000
Traite	Traite	Booléen			O	■	0
IdScénario	Idscénario	Entier sur 4 octets			O	■	0
idEquipement	Idequipement	Entier sur 4 octets			O	■	0
GPS	Gps	Chaîne	50				
Autre_Data	Autre_data	Chaîne	100				
IP_Adresse	Ip_adresse	Chaîne	50		O	■	
URL_EPCIS	Url_epcis	Chaîne	50				

ReceivePDA

Fichiers et rubriques

Informations générales

ReceivePDA	ReceivePDA
Nom sur disque	ReceivePDA.fic

Rubriques du fichier ReceivePDA

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
IDReceive	Identifiant de Receive	Identifiant automatique (8 octets)		O		■	
Mode	Mode	Entier non signé sur 1 octet					0
Data_In	Data_in	Chaîne	50				
Data_Out	Data_out	Chaîne	50				
TimùeStamp	Timùestamp	Date et Heure (aaaammjjhhmmssccc)					
Traite	Traite	Booléen			O	■	0
IdScénario	Idscénario	Entier sur 4 octets			O	■	0
idEquipement	Idequipement	Entier sur 4 octets			O	■	0
GPS	Gps	Chaîne	50				
Autre_Data	Autre_data	Chaîne	100				
IP_Adresse	Ip_adresse	Chaîne	50		O	■	
URL_EPCIS	Url_epcis	Chaîne	50				

scénario

Fichiers et rubriques

Informations générales

scénario	scénario (partagé)
Connexion	Connexion à '127.0.0.1'

Rubriques du fichier scénario

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
idScénario	idScénario	Identifiant automatique (4 octets)		O		■	
Name	Name	Chaîne unicode	45				
Comment	Comment	Image (mémo binaire)					

srv_epcis

Fichiers et rubriques

Informations générales

srv_epcis	srv_epcis (partagé)
Connexion	Connexion à '127.0.0.1'

Rubriques du fichier srv_epcis

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
idSrv_EPCIS	idSrv_EPCIS	Identifiant automatique (4 octets)		O		■	
IP_Address	IP_Address	Chaîne unicode	45		O	■	
dns	dns	Chaîne unicode	100		O	■	

srv_epcis1

Fichiers et rubriques

Informations générales

srv_epcis1	srv_epcis (partagé)

Nom sur disque	srv_epcis1.fic
Connexion	CnxEcolog

Rubriques du fichier srv_epcis1

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
idSrv_EPCIS	idSrv_EPCIS	Identifiant automatique (4 octets)		○		■	
IP_Address	IP_Address	Chaîne unicode	45		○	■	
dns	dns	Chaîne unicode	100		○	■	

Partie 4

Fenêtre WinDev

FEN_test

Code

Déclarations globales de FEN_test

```
PROCEDURE MaFenêtre()
```

FEN_test

Code des champs

Clic sur BTN_SansNom1

```
cd est une chaîne ="00806141411234567896"
Traitement_Msg()
```

Partie 5

Collection de procédures

COL_ProcéduresGlobales

Code

Procédure globale AI_Value__Extract

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//[ <Résultat> = ] AI_Value__Extract (<AI>, <Code128>)
//
// Paramètres :
//  AI : <indiquez ici le rôle de AI>
//  Code128 : <indiquez ici le rôle de Code128>
// Valeur de retour :
//  chaîne ANSI : //    Aucune
//
// Exemple :
// Indiquez ici un exemple d'utilisation.
//
PROCEDURE AI_Value__Extract(AI,Code128)

sValRet est une chaîne = ""
sAi est une chaîne = ai
TANTQUE Taille (sai) < 2
    sai ="0"+sai
FIN

sAi  = "("+sai+")"


c est un caractère
P est un entier
bSortie est un boolean = False

P = Position(Code128,sai)
IF P > 0 THEN
    P+=4
    TANTQUE PAS bSortie
        c=Milieu(Code128,P,1)
        IF Asc(c) >47 AND Asc(c) <(58) THEN
            svalret += c
            P++
        ELSE
            bSortie =
```

```
True
        END
    END
END

RENVOYER(svalret)
```

Procédure globale Base_to_Decimal__Convert_

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//[ <Résultat> = ] Base_to_Decimal__Convert_ (<Base>, <value>)
//
// Paramètres :
//  Base : <indiquez ici le rôle de Base>
//  value : <indiquez ici le rôle de value>
// Valeur de retour :
//  entier sur 8 octets : //    Aucune
//
// Exemple :
// Indiquez ici un exemple d'utilisation.
//
PROCEDURE Base_to_Decimal__Convert_(Base , value)

nValret est un entier sur 8 bytes
nRes est un entier = value
nExp est un entier =0
Ch est un entier
t est un entier = Taille(value)

TANTQUE t > 0

    SELON Base
        CAS 2,8
            Ch = NumériqueVersChaîne(Milieu(value,t,1))
        CAS 16
            SELON Majuscule(NumériqueVersChaîne(Milieu(value,t,1)))
                CAS "0"
                    ch = 0
                CAS "1"
                    Ch = 1
                CAS "2"
                    Ch = 2
                CAS "3"
                    Ch = 3
                CAS "4"
                    Ch = 4
                CAS "5"
```

```
                Ch = 5
            CAS "6"
                Ch = 6
            CAS "7"
                Ch = 7
            CAS "8"
                Ch = 8
            CAS "9"
                Ch = 9
            CAS "A"
                Ch = 10
            CAS "B"
                Ch = 11
            CAS "C"
                Ch = 12
            CAS "D"
                Ch = 13
            CAS "E"
                Ch = 14
            CAS "F"
                Ch = 15

            AUTRE CAS

        FIN
    AUTRE CAS

FIN

nvalret += ((Puissance(base,nexp)) * ch)
t--
nexp++
END

RENVOYER(nvalret)
```

Procédure globale Caract_Delete_

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//[ <Résultat> = ] Caract_Delete_ (<str>, <car>)
//
// Paramètres :
```

```
//  str : <indiquez ici le rôle de str>
//  car : <indiquez ici le rôle de car>
// Valeur de retour :
//  chaîne ANSI : //    Aucune
//
// Exemple :
// Indiquez ici un exemple d'utilisation.
//
PROCEDURE Caract_Delete_(str,car)

sValret est une chaîne = ""
i est un entier

FOR i = 1 TO Taille(str)
    IF Milieu(str,i,1) <> car THEN
        svalret+= Milieu (str,i,1)
    END
END
RENVOYER (svalret)
```

Procédure globale Clear_Alien

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//Clear_Alien ()
//
// 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 1 seconde
// Elle sera répétée en boucle, en attendant 10 minutes entre chaque appel
// Chaque appel suivant exécute une seule fois la procédure, sans timer
//

PROCEDURE Clear_Alien()
HSupprimeTout (Alien_Rfid)
```

Procédure globale Code128_To_GRAI96__Convert

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//[ <Résultat> = ] Code128_To_GRAI96__Convert (<cb>)
//
// Paramètres :
//  cb : <indiquez ici le rôle de cb>
// Valeur de retour :
//  chaîne ANSI : //    Aucune
//
// Exemple :
// Indiquez ici un exemple d'utilisation.
//
PROCEDURE Code128_To_GRAI96__Convert(cb)

sValRet est une chaîne = ""

sUrn1 est une chaîne = "urn:epc:id:grai:"
sUrn2 est une chaîne = "urn:epc:tag:grai-96:3."

sVal1 sont des chaînes
Tprefix est un entier = Prefix_Company__Size (Milieu(cb,6,12))

sval1 = Milieu(cb,6,tprefix)+"."+Milieu(cb,6+tprefix,5)+"."+SansEspace(Milieu(cb,12+tprefix,20))
sValRet = surn1+sval1


RENVOYER(svalret)
```

Procédure globale Code128_To_SGTIN96__Convert

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//[ <Résultat> = ] Code128_To_SGTIN96__Convert (<cb>)
//
// Paramètres :
//  cb : <indiquez ici le rôle de cb>
// Valeur de retour :
//  chaîne ANSI : //    Aucune
//
// Exemple :
// Indiquez ici un exemple d'utilisation.
//
PROCEDURE Code128_To_SGTIN96__Convert(cb)

sValRet est une chaîne = ""
urn1 est une chaîne = "urn:epc:id:sgtin:"
```

```
urn2 est une chaîne = "urn:epc:tag:sgtin-96:3."
nHeader est un entier = 48
nFilter est un entier = 3
nPartition est un entier = 5
sCompany est une chaîne
sIndicator est une chaîne
sSerial est une chaîne



sVal1,sVal2,sVal3,sVal4,sVal5 sont des chaînes
p est un entier = (Position (cb,"21",17))-1
cb = "("+Gauche(cb,2)+")"+Milieu(cb,3,p-2)+"(21)"+Milieu(cb,p+3,10)
sGtin est une chaîne =AI_Value__Extract(01,cb)
Tsgtin est un entier = Taille(sgtin)
sSerialNumber est une chaîne = NumériqueVersChaîne(Val(AI_Value__Extract(21,cb)))
sVal1 = urn1 + Milieu(sGtin,2,7)+"."+Gauche(sGtin,1)+Milieu(sGtin,9,Tsgtin-9)+"."+sSerialNumber
sVal2 = urn2 + Milieu(sGtin,2,7)+"."+Gauche(sGtin,1)+Milieu(sGtin,9,Tsgtin-9)+"."+sSerialNumber
scompany = Decimal_to_Base__Convert(2,Val(ExtraitChaîne(ExtraitChaîne(sval1,5,":"),1,".")),24)
sIndicator = Decimal_to_Base__Convert(2,Val(ExtraitChaîne(sVal1,2,".")),20)
sSerial = Decimal_to_Base__Convert(2,Val(ExtraitChaîne(sVal1,3,".")),38)
sval3 = Decimal_to_Base__Convert(2,nheader,8)+ Decimal_to_Base__Convert(2,nfilter,3)+ Decimal_to_Base__Convert(2,npartition,3)+ scompany
+ sindicator+sserial



svalret = sval1+";"+sval2+";"+sval3+";"+sval4+";"+sval5

RENVOYER (svalret)
```

Procédure globale Code128_To_SSCC96__Convert

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//[ <Résultat> = ] Code128_To_SSCC96__Convert (<cb>)
//
// Paramètres :
//  cb : <indiquez ici le rôle de cb>
// Valeur de retour :
//  chaîne ANSI : //    Aucune
//
// Exemple :
// Indiquez ici un exemple d'utilisation.
//
PROCEDURE Code128_To_SSCC96__Convert(cb)

sValRet est une chaîne = ""
```

```
sUrn1 est une chaîne = "urn:epc:id:sscc:"
sUrn2 est une chaîne = "urn:epc:tag:sscc-96:3."

sVal1 sont des chaînes
Tprefix est un entier = Prefix_Company__Size(Milieu(cb,4,12))

sval1 = Milieu(cb,4,tprefix)+"."+Milieu(cb,3,1)+Milieu(cb,4+tprefix,20-(tprefix+4))
sValRet = surn1+sval1


RENVOYER(svalret)
```

Procédure globale Code_Hexa_To_SGTIN96__Convert

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//[ <Résultat> = ] Code_Hexa_To_SGTIN96__Convert (<code>)
//
// Paramètres :
//  code : <indiquez ici le rôle de code>
// Valeur de retour :
//  chaîne ANSI : //    Aucune
//
// Exemple :
// Indiquez ici un exemple d'utilisation.
//
PROCEDURE Code_Hexa_To_SGTIN96__Convert(code)

sValRet est une chaîne = "urn:epc:id:sgtin:"
sValBin est une chaîne

svalbin = Hexa_to_Binaire__Convert(code)
sValRet += NumériqueVersChaîne(Base_to_Decimal__Convert(2,Milieu(svalbin,15,24)), "07d")
svalret +="."+NumériqueVersChaîne(Base_to_Decimal__Convert(2,Milieu(sValBin,39,20)), "06d")
svalret +="."+Base_to_Decimal__Convert_(2,Milieu(sValBin,59,38))
RENVOYER (sValRet)
```

Procédure globale Code_Hexa_To_SSCC96__Convert

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//[ <Résultat> = ] Code_Hexa_To_SSCC96__Convert (<Code>)
//
// Paramètres :
//  Code : <indiquez ici le rôle de Code>
// Valeur de retour :
```

```
//  chaîne ANSI : //    Aucune
//
// Exemple :
// Indiquez ici un exemple d'utilisation.
//
PROCEDURE Code_Hexa_To_SSCC96__Convert(Code)


sValRet est une chaîne = "urn:epc:id:sscc:"
sValBin est une chaîne

sValBin = Hexa_to_Binaire__Convert(code)
sValRet += NumériqueVersChaîne(Base_to_Decimal__Convert(2,Milieu(sValBin,15,24)), "07d")
sValRet +="."+Base_to_Decimal__Convert_(2,Milieu(sValBin,39,34))


RENVOYER (sValRet)
```

Procédure globale FinPgm

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//FinPgm ()
//
// Paramètres :
//  Aucun
// Valeur de retour :
//  Aucune
//
// Exemple :
// Indiquez ici un exemple d'utilisation.
//// Traitement automatique des erreurs : RENVOYER
//

PROCEDURE FinPgm()
ExeTermine(ExeDonnePID())
```

Procédure globale 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)


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

RENVOYER(svalret)
```

Procédure globale Lecture_Alien

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//Lecture_Alien ()
//
// Paramètres :
//  Aucun
// Valeur de retour :
//  Aucune
//
// Exemple :
// Indiquez ici un exemple d'utilisation.
//// Traitement automatique des erreurs :   exécuter le bloc de code CAS ERREUR
//
// Traitement automatique des exceptions :   appeler la procédure 'FinPgm'
//
// Procédure automatique :
// La procédure est exécutée automatiquement, après le code d'initialisation du projet, avec un différé de 10 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 10 secondes entre chaque appel
//

PROCEDURE Lecture_Alien()

SectionCritiqueDébut("ALIEN")
bPrompt is boolean = False
Resultat est une chaîne
sLigne est une chaîne
```

```
sligneGen est une chaîne
i est un entier

HFiltre(equipements,Equipement_Type,3)
HLitPremier (equipements,Equipement_Type)

TANTQUE PAS HEnDehors
    gsAlien_Ip=equipements.IP_Address
    gsAlien_Login="alien"
    gsAlien_Paswword="password"
    gsALien_FileNameResut="ResultAlien1.txt)"
    gnAlien_RF_Attenuation=equipements.Rfid_Attenuation
    IF Ping(gsAlien_Ip,1000)  THEN

        gclMyClsReader:InitOnNetwork (gsAlien_Ip,23)
        gclMyClsReader:Connect()
        IF gclMyClsReader:Login (gsAlien_Login, gsAlien_Paswword) THEN
            gclMyClsReader:set_RFAttenuation (gnAlien_RF_Attenuation)
            Resultat = gclMyClsReader:SendReceive("get TagList",bPrompt)
        END
        gclMyClsReader:Disconnect()

    END
    i = 1
    bRFID_Present est un boolean

    sligneGen = resultat
    HRAZ(Receive)
    sligne = ExtraitChaîne(slignegen,i,RC)
    TANTQUE SansEspace(sLigne) <> EOT
        i++
        bRFID_Present = False
        gsMsgRet= Milieu(sligne,5,30)
        gsMsgRet = Caract_Delete_(gsMsgRet,"")
        gsMsgRet = Caract_Delete_(gsMsgRet,",")

        HLitRecherche(Alien_Rfid,NumFabr,gsMsgRet)
        IF PAS HTrouve THEN
            IF (Val(gsMsgRet) > 0)  THEN
                HRAZ(Alien_Rfid)
                Alien_Rfid.NumFabr=gsMsgRet
                Alien_Rfid.DATE =Today()+Now()
                Alien_Rfid.Traite=1
                Alien_Rfid.IP_Adresse = gsAlien_Ip
                HAjoute(Alien_Rfid)
                //Hraz(Receive)
                Receive.Data_In+=Alien_Rfid.NumFabr+";"
//              Receive.TimùeStamp = Alien_Rfid.DATE
//              Receive.Mode = 2
```

```
//              receive.traite =0
//              receive.IP_Adresse= Alien_Rfid.IP_Adresse
//              Receive.URL_EPCIs= url__epcis(Alien_Rfid.IP_Adresse)
                //hajoute(Receive)

            END
        END
        sligne = ExtraitChaîne(slignegen,i,RC)
    FIN
    Receive.TimùeStamp = Alien_Rfid.DATE
    Receive.Mode = 2
    Receive.Traite =0
    Receive.IP_Adresse= Alien_Rfid.IP_Adresse
    Receive.URL_EPCIS= URL__EPCIS(Alien_Rfid.IP_Adresse)
    IF SansEspace(Receive.Data_In) <> "" THEN HAjoute(Receive)
    HLitSuivant(equipements,Equipement_Type)
END
HDésactiveFiltre(equipements)
SectionCritiqueFin ("ALIEN")

CAS ERREUR:
```

Procédure globale Logger

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//Logger (<ind>, <txt>)
//
// Paramètres :
//  ind : <indiquez ici le rôle de ind>
//  txt : <indiquez ici le rôle de txt>
// Valeur de retour :
//  Aucune
//
// Exemple :
// Indiquez ici un exemple d'utilisation.
//
PROCEDURE Logger(ind, txt )

sligne est une chaîne

nFic est un entier = fOuvre("log.txt",foCréationSiInexistant+foLectureEcriture)
IF nfic > 0 THEN
    IF ind = 1 THEN
        sligne = "OK" + DateVersChaîne(Today) + ""+ HeureVersChaîne(Now,"HH:MM:SS") + txt
    ELSE
        sligne = "KO" + DateVersChaîne(Today) + ""+ HeureVersChaîne(Now,"HH:MM:SS") + txt
    END
    fEcritLigne(nfic,sligne)
```

```
    fFerme(nfic)
END
```

Procédure globale Main_Code128_TO_EPCGen2__Convert

```
PROCEDURE Main_Code128_TO_EPCGen2__Convert(Code128)

sValRet est une chaîne = ""


IF Position(Code128,"00") = 1 THEN sValRet = Code128_To_SSCC96__Convert(code128)
IF Position(Code128,"01") = 1 THEN sValRet = ExtraitChaîne(Code128_To_SGTIN96__Convert(code128),1,";")
IF Position(Code128,"8003") = 1 THEN sValRet = Code128_To_GRAI96__Convert(Code128)




RENVOYER(svalret)
```

Procédure globale Main_HEXA_TO_EPCGen2__Convert

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//[ <Résultat> = ] Main_HEXA_TO_EPCGen2__Convert (<CodeHexa>)
//
// Paramètres :
//  CodeHexa : <indiquez ici le rôle de CodeHexa>
// Valeur de retour :
//  chaîne ANSI : //    Aucune
//
// Exemple :
// Indiquez ici un exemple d'utilisation.
//
```

```
PROCEDURE Main_HEXA_TO_EPCGen2__Convert (CodeHexa)

sValRet est une chaîne = ""

IF Position(Codehexa,"31") > 0 THEN sValRet = Code_Hexa_To_SSCC96__Convert(Codehexa)
IF Position(Codehexa,"30") > 0 THEN sValRet = Code_Hexa_To_SGTIN96__Convert(codeHexa)

//END

RENVOYER(svalret)
```

Procédure globale Prefix_Company__Size

```
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
//[ <Résultat> = ] Prefix_Company__Size (<Code>)
//
// Paramètres :
//  Code : <indiquez ici le rôle de Code>
// Valeur de retour :
//  entier : //     Aucune
//
// Exemple :
// Indiquez ici un exemple d'utilisation.
//
PROCEDURE Prefix_Company__Size(Code)

nValRet est un entier = 7
bSortie est un boolean = False
sCle est une chaîne
i est un entier = Taille(code)

TANTQUE PAS bSortie AND (i > 0)
    scle = Gauche(code,i)
    HLitRecherche(prefix,Prefix_Code,scle)
    IF HTrouve THEN
        nvalret = prefix.Prefix_Length
        bsortie = True
    END
    i--
FIN

RENVOYER (nvalret)
```

Procédure globale Traitement_Msg

```
PROCEDURE Traitement_Msg()

i est une entier = 1
ideqts, sUrl est une chaîne
sCleIn sont des chaîne
HDésactiveFiltre(Receive)
HFiltre(Receive,Traite,0)
HLitPremier(Receive,Traite)
TANTQUE PAS HEnDehors
    //if SansEspace(receive.Data_Out) = "" then
    SELON Receive.Mode
        CAS 1 // 128

            sCleIn = ExtraitChaîne(Receive.Data_In,1,";")
            TANTQUE Taille (sCleIn) > 10
                IF Position(sCleIn,RC) > 0 THEN
                    sCleIn = Milieu(sCleIn,3,Taille(sCleIn)-2)
                END
                Receive.Data_Out += Main_Code128_TO_EPCGen2__Convert(sCleIn)+";"
                i ++
                sCleIn = ExtraitChaîne(Receive.Data_In,i,";")
            END

        CAS 2 // HEXA
            //receive.Data_Out= Main_HEXA_TO_EPCGen2__Convert (receive.Data_In)

            sCleIn = ExtraitChaîne(Receive.Data_In,i,";")
            TANTQUE Taille (sCleIn) > 10
                IF Position(sCleIn,RC) > 0 THEN
                    sCleIn = Milieu(sCleIn,3,Taille(sCleIn)-2)
                END
                Receive.Data_Out += Main_HEXA_TO_EPCGen2__Convert(sCleIn)+";"
                i++
                sCleIn = ExtraitChaîne(Receive.Data_In,i,";")
            END
        CAS 3 // URI
            Receive.Data_Out = Receive.Data_In

        AUTRE CAS

    FIN
    Receive.Traite=1
    HModifie(Receive)

    //end
    ideqts = Receive.IP_Adresse
    IF SansEspace(ideqts)="" THEN ideqts = Key_Eqt__Compute(Receive.IdScénario,Receive.idEquipement)
    HLitRecherche(equipements,IP_Address,ideqts)
    IF HTrouve THEN
```

```
        sUrl = equipements.EPCIS_URL_Capture
        IF XML__Compute(ideqts,sUrl) THEN
            Receive.URL_EPCIS = sUrl
            HModifie(Receive)
            DélaiAvantFermeture(150)
            Info("ok")
            DélaiAvantFermeture()
        END
    END
    HLitSuivant(Receive,Traite)
FIN
HDésactiveFiltre(Receive)
```

Procédure globale URL__EPCIS

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

sValRet est une chaîne ="http://localhost:8080/epcis-repository-0.5.0/capture"

HLitRecherche(equipements,IP_Address,IP)
IF HTrouve THEN
    HLitRecherche(srv_epcis1,idSrv_EPCIS,equipements.idEPCIS)
    IF HTrouve THEN
        sValRet="http://"+srv_epcis1.IP_Address+":8080/epcis-repository-0.5.0/capture"
    END
END
RENVOYER (sValRet)
```

Procédure globale XML__Compute

```
PROCEDURE XML__Compute(idEqt,sUrl)

i est un entier = 1
svalue est une chaîne
```

```
sLigne est une chaîne
ideqt=""
bvalret est un boolean = True
nFic est un entier
//*****************************************************************************************************************
//
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)
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)
Svalue = ExtraitChaîne(Receive.Data_Out,i,";")
TANTQUE Taille (svalue) > 20
    sLigne += "<epc>"+svalue+"</epc>";
    i++//FecritLigne(nFic,sLigne)
    svalue = ExtraitChaîne(Receive.Data_Out,i,";")
END
sLigne += "</epcList>" ;//FecritLigne(nFic,sLigne)
sligne += "<action>"+equipements.Action+"</action>"; //FecritLigne(nFic,sLigne)
sLigne += "<bizStep>"+equipements.BizStepName+"</bizStep>"; //FecritLigne(nFic,sLigne)
sLigne += "<disposition>"+equipements.BizDispositionName+"</disposition>"; //FecritLigne(nFic,sLigne)
sLigne += "<readPoint>"; //FecritLigne(nFic,sLigne)
sLigne += "<id>"+equipements.ReadPointName+"</id>"; //FecritLigne(nFic,sLigne)
sLigne += "</readPoint>"; //FecritLigne(nFic,sLigne)
sLigne += "<bizLocation>"; //FecritLigne(nFic,sLigne)
sLigne += "<id>"+equipements.BizLocName+"</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)
//*****************************************************************
HTTPCréeFormulaire("Form")
HTTPAjouteParamètre("Form","",sligne)
IF HTTPEnvoieFormulaire("Form",sUrl,httpPost, "", "", "text/xml")  THEN
    bvalret=1
END
IF bvalret = 1 THEN
    repname est une chaîne = ("C:\Data")
    fRepCrée(repname)
    sFilename est une chaîne = ComplèteRep(repname) + "Post.log"
    nfic = fOuvre (sFilename,foCréationSiInexistant+foLectureEcriture+foAjout)
    IF nfic > 0 THEN
```

```
        fEcritLigne(nfic,sligne)
        fFerme(nFic)
    END

END
//*****************************************************************

//end
RENVOYER(bvalret)
```

Procédure globale XML__Post

```
PROCEDURE XML__Post(msg)

bOK est un boolean = False

HTTPCréeFormulaire("Form")
HTTPAjouteParamètre("Form","",msg)
IF HTTPEnvoieFormulaire("Form",gsPathEPCIS_Server,httpPost, "", "", "text/xml") THEN
    bok = True
ELSE
    bok = False
END
RENVOYER(bok)
```

Partie 6

Table des matières

Table des matières

Projet SV_Ecolog