

# Dossier de développement

## Documentation du Projet Winnix Mobile

04/05/2016

Application réalisée par

tel :  
fax :

## Partie 1

**En-tête**

# Documentation Winnix Mobile






## Partie 2

### Projet

**Projet****Informations générales**

winnix mobile serveur	
Répertoire	C:\Mes Projets\winnix mobile serveur
Première fenêtre	FEN_serveur
Analyse associée	C:\Mes Projets\winnix mobile serveur\winnix mobile serveur.ana\winnix mobile serveur.wda
Gabarit	190 Gensteel

**Projet****Liste des éléments du projet**

	Nom	Libellé
	FEN_serveur	
	REQ_Ajout_Visite	
	REQ_Modif_Visite	
	REQ_Visualisation	
	Aide_winnix_mobile_serveur	

Statistiques	
Nombre de fenêtres :	1
Nombre de requêtes :	3
Nombre de fichiers d'aide :	1

**Projet****Tableau de bord****Projet****Code****Initialisation de winnix mobile serveur**

GLOBAL

```
gsConnexion est une chaîne UNICODE = "connexion"
gsMotDePasse est une chaîne
```

**Projet****Erreurs, warnings et informations**

	Détail de l'erreur	Localisation	Ligne	Colon
❌	Aucun 'SORTIR', 'RETOUR' ou 'REVOYER' n'a été détecté dans le code de la boucle : il y a un risque de boucle sans fin.	FEN_serveur.PROCEDURE.Connexion Procédure locale	15	1
❌	Aucun 'SORTIR', 'RETOUR' ou 'REVOYER' n'a été détecté dans le code de la boucle : il y a un risque de boucle sans fin.	FEN_serveur.PROCEDURE.actualiseMDP Procédure locale	14	1

**Projet****Statistiques sur le code**

Lignes	%	Lig./tr	
258	44	23	FEN_serveur
12	0	12	winnix mobile serveur
<b>261</b>	<b>44</b>	<b>21</b>	

**Lignes:** Nombre total de lignes de code.

**% comm.:** Pourcentage de commentaires dans le code.

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

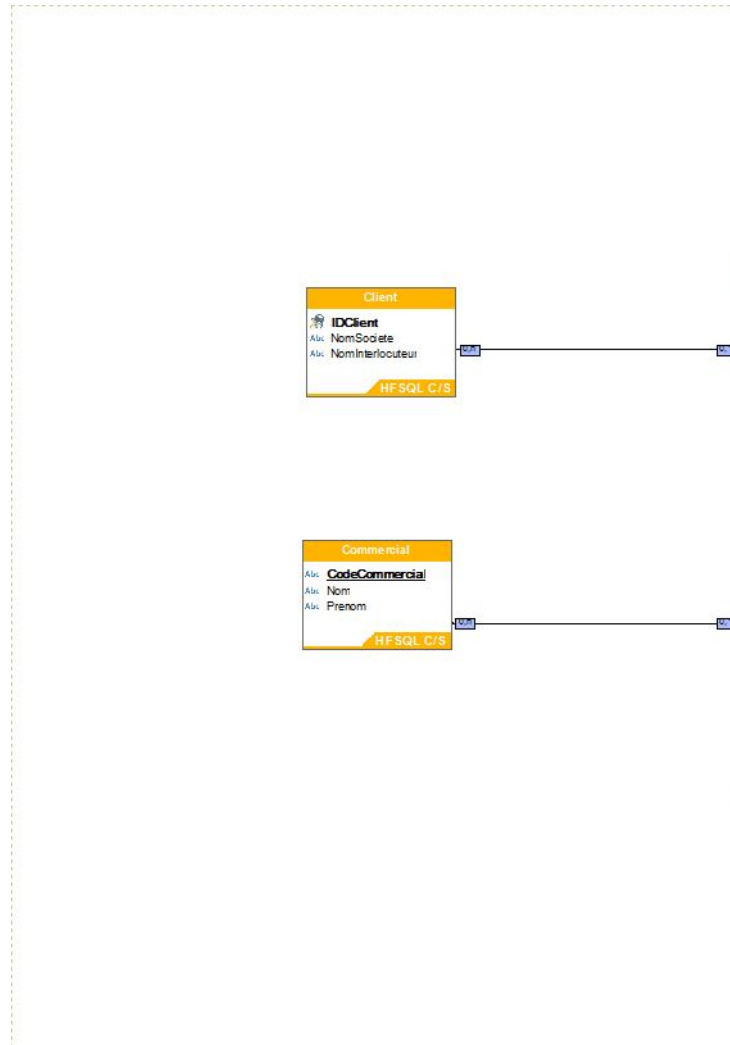
## Partie 3

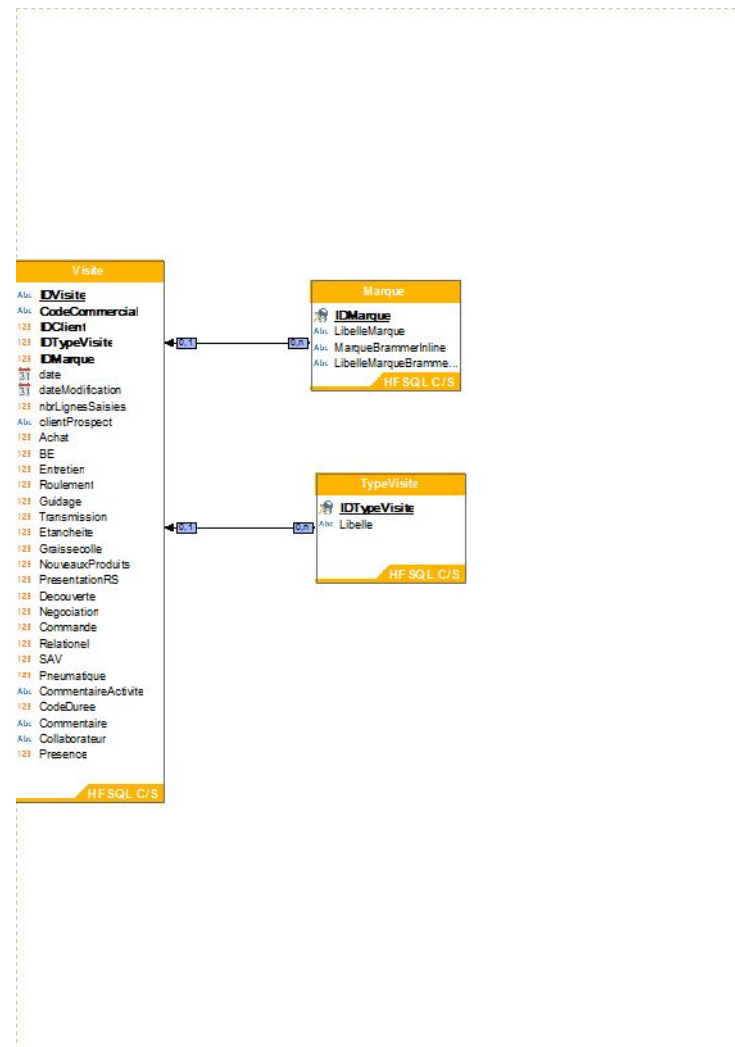
### Analyse



## Analyse

## Graphe





## Analyse

### Informations générales

winnix mobile serveur.wda	
Répertoire	C:\Mes Projets\winnix mobile serveur\winnix mobile serveur.ana\
Répertoire des fichiers de données	Répertoire de l'application

N°génération	Nb. fichiers	Nb. rubriques	Nb. liaisons	Nb. connexions	Nb. groupes
8	5	38	4	1	0

## Analyse

### Dictionnaire des rubriques

Rubrique	Type	Taille	Clé	Clé	Utilisée par...
Achat	Booléen				Visite
BE	Booléen				Visite
clientProspect	Caractère				Visite
CodeCommercial	Chaîne	50	○		Commercial
	Chaîne	50		○	Visite
CodeDuree	Entier sur 4 octets				Visite
Collaborateur	Chaîne	50			Visite
Commande	Booléen				Visite
Commentaire	Chaîne	50			Visite
CommentaireActivite	Chaîne	50			Visite
date	Date (aaaammjj)				Visite
dateModification	Date (aaaammjj)				Visite
Decouverte	Booléen				Visite
Entretien	Booléen				Visite

Rubrique	Type	Taille	Clé	Clé	Utilisée par...
Etancheite	Booléen				Visite
Graissecolle	Booléen				Visite
Guidage	Booléen				Visite
IDClient	Identifiant automatique (4 octets)		○		Client
	Entier sur 4 octets			○	Visite
IDMarque	Identifiant automatique (8 octets)		○		Marque
	Entier sur 8 octets			○	Visite
IDTypeVisite	Identifiant automatique (8 octets)		○		TypeVisite
	Entier sur 8 octets			○	Visite
IDVisite	Chaîne	50	○		Visite
Libelle	Chaîne	50			TypeVisite
LibelleMarque	Chaîne	50			Marque
LibelleMarqueBrammerInline	Chaîne	50			Marque
MarqueBrammerInline	Chaîne	50			Marque
nbrLignesSaisies	Entier sur 4 octets				Visite
Negociation	Booléen				Visite
Nom	Chaîne	50			Commercial
NomInterlocuteur	Chaîne	50			Client
NomSociete	Chaîne	50			Client
NouveauxProduits	Booléen				Visite
Pneumatique	Booléen				Visite
Prenom	Chaîne	50			Commercial
Presence	Entier sur 4 octets				Visite
PresentationRS	Booléen				Visite
Relationel	Booléen				Visite
Roulement	Booléen				Visite
SAV	Booléen				Visite
Transmission	Booléen				Visite

**Analyse****Liaisons**

<b>Contrainte_Client_Visite</b>	
---------------------------------	--

	Source	Relié
Fichier	Client	Visite
Rubrique	IDClient (<Mémo binaire inconnu>)	IDClient (<Mémo binaire inconnu>)
Cardinalités	0,n	0,1

Modification	Modification uniquement si pas d'enregistrement relié
Suppression	Suppression uniquement si pas d'enregistrement relié

<b>Contrainte_Commercial_Visite</b>	
-------------------------------------	--

	Source	Relié
Fichier	Commercial	Visite
Rubrique	CodeCommercial (<Mémo binaire inconnu>)	CodeCommercial (<Mémo binaire inconnu>)
Cardinalités	0,n	0,1

Modification	Modification uniquement si pas d'enregistrement relié
Suppression	Suppression uniquement si pas d'enregistrement relié

<b>Contrainte_Marque_Visite</b>	
---------------------------------	--

Source	Relié
--------	-------

	Source	Relié
Fichier	Marque	Visite
Rubrique	IDMarque (<Mémo binaire inconnu>)	IDMarque (<Mémo binaire inconnu>)
Cardinalités	0,n	0,1

Modification	Modification uniquement si pas d'enregistrement relié
Suppression	Suppression uniquement si pas d'enregistrement relié

Contrainte\_TypeVisite\_Visite

	Source	Relié
Fichier	TypeVisite	Visite
Rubrique	IDTypeVisite (<Mémo binaire inconnu>)	IDTypeVisite (<Mémo binaire inconnu>)
Cardinalités	0,n	0,1

Modification	Modification uniquement si pas d'enregistrement relié
Suppression	Suppression uniquement si pas d'enregistrement relié

## Analyse

### Connexions

Connexion	Source de données	Utilisateur	Provider	Tm.	Tm.
<b>MaConnexion</b> Connexion à 'PCFR-EU0015'	PCFR-EU0015:4900	admin	WinDevClientServeurHF	30	30
Utilisée par...	Client TypeVisite	Commercial Visite	Marque		

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
<b>Client</b>		○	36	115	○						○	○	○		HFSQL Client/Serveur
<b>Commercial</b>			4	162	○						○	○	○		HFSQL Client/Serveur
<b>Marque</b>		○	4	170	○						○	○	○		HFSQL Client/Serveur
<b>TypeVisite</b>		○	1	68	○						○	○	○		HFSQL Client/Serveur
<b>Visite</b>			9	334	○						○	○	○		HFSQL Client/Serveur

Espaces \*: Complétion des chaînes par des espaces

Jnl. Lect/Ecr \*: Journalisation des lectures et écritures

Jnl. Ecr \*: Journalisation des écritures

Sécurité \*: Mode sécurité renforcée

## Client

### Fichiers et rubriques

Informations générales

<b>Client</b>	Clients/Prospects
<b>Nom sur disque</b>	Client.FIC
<b>Connexion</b>	Connexion à 'PCFR-EU0015'

Rubriques du fichier Client

	<b>Libellé</b>	<b>Type</b>	<b>Taille</b>	<b>Clé</b>	<b>Clé</b>	<b>Sens</b>	<b>Val. défaut</b>
<b>IDClient</b>	Identifiant de Client	Identifiant automatique (4 octets)		○		■	
<b>NomSociete</b>	Nomsociete	Chaîne	50				
<b>NomInterlocuteur</b>	Nominterlocuteur	Chaîne	50				



## Commercial

### Fichiers et rubriques

#### Informations générales

<b>Commercial</b>	Commercial
<b>Nom sur disque</b>	Commercial.FIC
<b>Connexion</b>	Connexion à 'PCFR-EU0015'

#### Rubriques du fichier Commercial

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
<b>CodeCommercial</b>	Codecommercial	Chaîne	50	○		■	
<b>Nom</b>	Nom	Chaîne	50				
<b>Prenom</b>	Prenom	Chaîne	50				

## Marque

### Fichiers et rubriques

#### Informations générales

<b>Marque</b>	Marque
<b>Nom sur disque</b>	Marque.FIC
<b>Connexion</b>	Connexion à 'PCFR-EU0015'

#### Rubriques du fichier Marque

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
<b>IDMarque</b>	Identifiant de Marque	Identifiant automatique (8 octets)		○		■	
<b>LibelleMarque</b>	Libellemarque	Chaîne	50				
<b>MarqueBrammerInline</b>	Marquebrammerinline	Chaîne	50				
<b>LibelleMarqueBrammerInline</b>	Libellemarquebrammerinline	Chaîne	50				

## TypeVisite

### Fichiers et rubriques

#### Informations générales

<b>TypeVisite</b>	TypeVisite
<b>Nom sur disque</b>	TypeVisite.fic
<b>Connexion</b>	Connexion à 'PCFR-EU0015'

#### Rubriques du fichier TypeVisite

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
<b>IDTypeVisite</b>	Identifiant de TypeVisite	Identifiant automatique (8 octets)		○		■	
<b>Libelle</b>	Libelle	Chaîne	50				

## Visite

### Fichiers et rubriques

#### Informations générales

<b>Visite</b>	Visite
<b>Nom sur disque</b>	Visite.FIC
<b>Connexion</b>	Connexion à 'PCFR-EU0015'

#### Rubriques du fichier Visite

	Libellé	Type	Taille	Clé	Clé	Sens	Val. défaut
<b>IDVisite</b>	Identifiant de Visite	Chaîne	50	○		■	0
<b>date</b>	Date	Date (aaaammjj)					
<b>dateModification</b>	Datemodification	Date (aaaammjj)					
<b>nbrLignesSaisies</b>	Nbrlignessaisies	Entier sur 4 octets					0
<b>clientProspect</b>	Clientprospect	Caractère					
<b>Presence</b>	Presence	Entier sur 4 octets					1
<b>Achat</b>	Achat	Booléen					0
<b>BE</b>	Be	Booléen					0
<b>Entretien</b>	Entretien	Booléen					0
<b>Roulement</b>	Roulement	Booléen					0
<b>Guidage</b>	Guidage	Booléen					0
<b>Transmission</b>	Transmission	Booléen					0
<b>Etancheite</b>	Etancheite	Booléen					0
<b>Graissecolle</b>	Graissecolle	Booléen					0
<b>NouveauxProduits</b>	Nouveauxproduits	Booléen					0
<b>PresentationRS</b>	Presentationrs	Booléen					0
<b>Decouverte</b>	Decouverte	Booléen					0
<b>Negociation</b>	Negociation	Booléen					0
<b>Commande</b>	Commande	Booléen					0

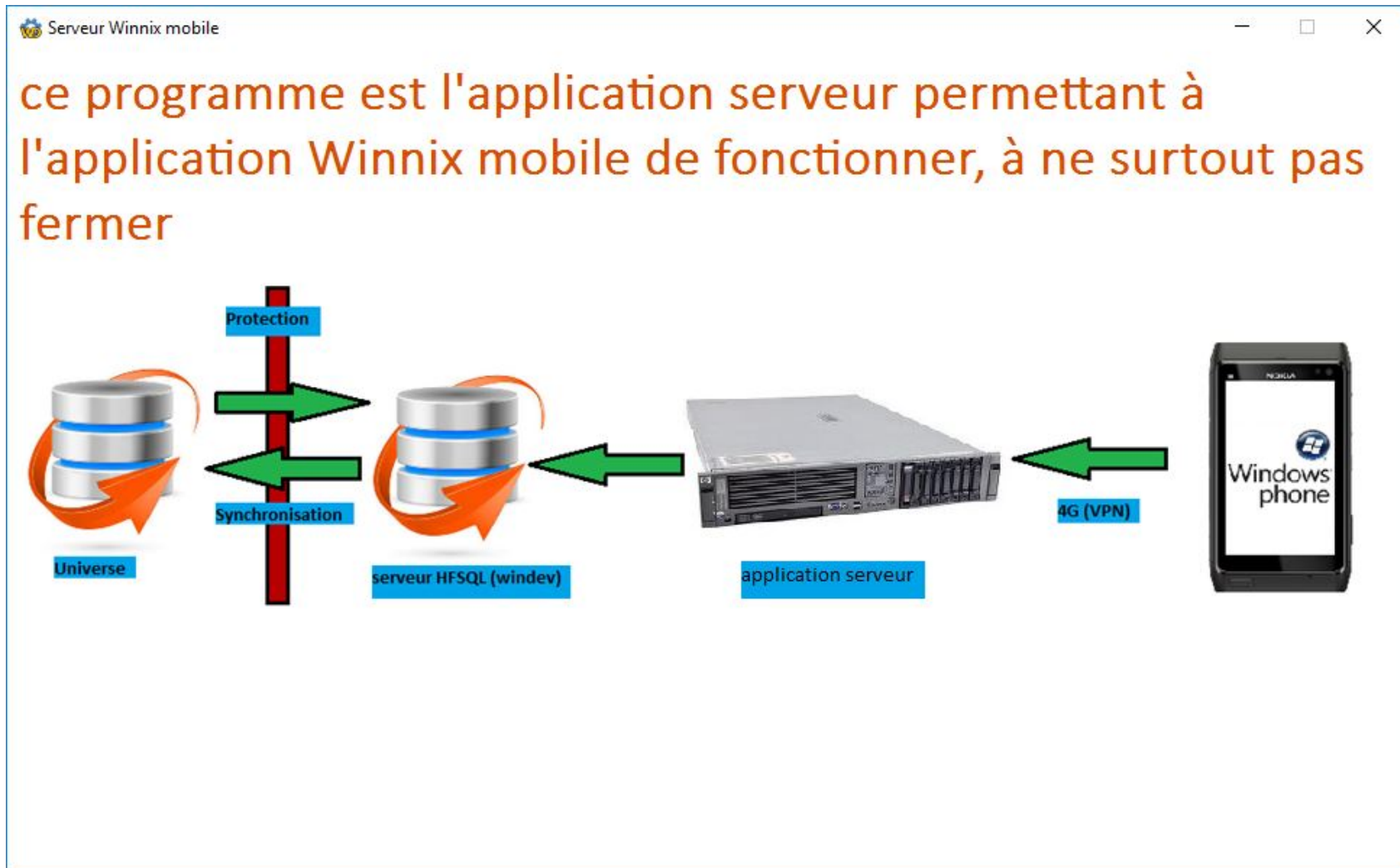
	<b>Libellé</b>	<b>Type</b>	<b>Taille</b>	<b>Clé</b>	<b>Clé</b>	<b>Sens</b>	<b>Val. défaut</b>
<b>Relationel</b>	Relationel	Booléen					0
<b>SAV</b>	Sav	Booléen					0
<b>Pneumatique</b>	Pneumatique	Booléen					0
<b>CommentaireActivite</b>	Commentaireactivite	Chaîne	50				
<b>CodeDuree</b>	Codeduree	Entier sur 4 octets					0
<b>Commentaire</b>	Commentaire	Chaîne	50				
<b>Collaborateur</b>	Collaborateur	Chaîne	50				
<b>IDClient</b>	IDClient	Entier sur 4 octets			○	■	0
<b>CodeCommercial</b>	CodeCommercial	Chaîne	50		○	■	
<b>IDMarque</b>	IDMarque	Entier sur 8 octets			○	■	0
<b>IDTypeVisite</b>	IDTypeVisite	Entier sur 8 octets			○	■	0

## Partie 4

### Fenêtre WinDev

## FEN\_serveur

Image



**FEN\_serveur****Informations générales**

<b>FEN_serveur</b>	
--------------------	--

<b>Fichier physique</b>	C:\Mes Projets\winnix mobile serveur\FEN_serveur.wdw
<b>Crée par</b>	ALEXIS GSTALTER
<b>Date de création</b>	22/04/2016 12:57:41
<b>Dernière modification par</b>	ALEXIS GSTALTER
<b>Date dernière modification</b>	04/05/2016 14:45:16
<b>Titre</b>	FR: Serveur Winnix mobile
<b>Type fenêtre</b>	Fenêtre libre
<b>Mode ouverture (Desktop)</b>	Centrée par rapport à la mère

Note pour le dossier	
FR:	
FR:	

Position X	0	Position Y	0	Largeur	970
Hauteur	600				

Visible	<input checked="" type="radio"/>	Barre de titre	<input checked="" type="radio"/>	Menu système	<input checked="" type="radio"/>
Icône réduction	<input checked="" type="radio"/>	Icône agrandissement	<input type="radio"/>	Icône aide ctx.	<input type="radio"/>
Barre de message	<input type="radio"/>	Détourer le fond	<input type="radio"/>	Mode 9 images	<input type="radio"/>
Redimensionnable	<input type="radio"/>	Avec grip	<input checked="" type="radio"/>	Grip dans image	<input type="radio"/>
Déplace/fond	<input type="radio"/>	Sauve position	<input checked="" type="radio"/>	Ascenseur auto.	<input type="radio"/>
Gestion grande police	<input checked="" type="radio"/>	Menu look XP	<input checked="" type="radio"/>	Thème XP	<input type="radio"/>
Ctx. HFSQL indép.	<input type="radio"/>		<input type="radio"/>		<input type="radio"/>

Etat initial	Actif
Clic droit	
Curseur survol	0



Menu contextuel	
Fichier aide ctx.	
Opacité	100
Type cadre	Double
Transparence Vista	Non
Transparence gérée par Vista	Non
Valide auto.	0
Bouton auto.	
Lib. + tps restant	Oui
Groupe MDI	
Pos. X Menu	6
Pos. Y Menu	28
Mode menu ctx.	<Aucun>
M.C.U.	Oui
Anim. Ouverture	0
Anim. Fermeture	0
Durée Anim. Ouverture	100
Durée Anim. Fermeture	50
Adapte Taille contenu	Non
Assombri quand inactif	Oui
Déplace souris sur bouton par défaut	Non
Compact. barres outils	Non
Fichier parcouru	
Rub. parcourue	
Image	
Image fond	FR: C:\Mes Projets\winnix mobile serveur\schéma winnix mobile.png
Icône	
Image poignée	C:\Mes Projets\winnix mobile serveur\Gensteel_Resize.gif


**Statistiques de la fenêtre**

	Libellé	1		Bouton	1
---	---------	---	---	--------	---



Nombre d'éléments dans la fenêtre : 2

**FEN\_serveur****Informations sur les champs****Fenêtre : FEN\_serveur**

## ○ Libellé

	Position	Position Y	Largeur	Hauteur	Plan	Visib	Etat initial	Cadrage horz.	Cadrage vert.	Form	Anim
LIB_Bannière	6	6	956	146	0		Actif	Gauche	Haut		

## ○ Bouton

	Position	Position Y	Largeur	Hauteur	Plan	Visib	Etat initial	Touc	Type bouton
BTN_SansNom1	0	-85	48	47	0		Actif		Aide

**Fenêtre : FEN\_serveur**

○ Bouton : BTN\_SansNom1

**Note pour le dossier**

FR:
FR:

Libellé	FR: Aide
Ordre navigation	1
Curseur survol	0
IHM par l'utilisateur	Selon fenêtre
On/Off	Non
Recopie Option menu flèche	Non

Taux Paral. hauteur	0
Taux Paral. Y	0
Retrait gauche	0
Réduction Ruban	Ne pas réduire le bouton

Numéro aide	3
-------------	---

Opacité	100
---------	-----

Animée par défaut	Oui
Mode ellipse	<Aucun>
Type flèche	<Aucun>
Mode menu ctx.	<Aucun>

Nb. états image	1
Image fond	FR: C:\Mes Projets\winnix mobile serveur\Gensteel_Btn_Media_Right.png
Nb. états fond	5
Etirement de l'image de fond	Mode 9 images personnalisé
Nb. étapes animation	6
Largeur Halo	0
Hauteur Halo	0

Ancrage		<Aucun>					
Taux anc. droite	1000	Taux anc. bas	1000	Taux anc. largeur	1000	Taux anc. hauteur	1000

○ Libellé : LIB\_Bannière

**Note pour le dossier**

FR:
FR:

Libellé	FR: ce programme est l'application serveur permettant à l'application Winnix mobile de fonctionner, à ne surtout pas fermer
Curseur survol	0
Synchro Infos Partagées	Non
IHM par l'utilisateur	Selon fenêtre

Taux Paral. hauteur	0
Taux Paral. Y	0

Numéro aide	0
-------------	---

Opacité	100
---------	-----

Type saisie	Texte
Masque saisie	FR: 0
Mode ellipse	<Aucun>
Unicode	Non
Transforme le &	Non
Mise à blanc	Oui

Ancrage			<Aucun>							
Taux anc. droite	1000		Taux anc. bas	1000		Taux anc. largeur	1000		Taux anc. hauteur	1000

## FEN\_serveur

Code

### Déclarations globales de FEN\_serveur

```
PROCEDURE MaFenêtre()
```

### Fin d'initialisation de FEN\_serveur

```
SI SocketCrée("Serveur",8000) = Faux ALORS // sert aux échanges d'information au moment de la connexion
    Info("impossible de créer le serveur")
FIN
ThreadExécute("Connexion",threadCopieComplèteContexteHFSQL,Connexion)
// ce thread tourne en permanence, il reçoit des ordres de l'application cliente sous forme de mots
ThreadExécute("actualiseMDP",threadNormal,actualiseMDP)
//signalCrée("continu",signalAutomatique,signalOuvert)
```

## FEN\_serveur

### Code des champs

#### Clic sur BTN\_SansNom1

```
//// Ouverture de la page d'aide associée à la fenêtre en cours  
wHelp( "Aide winnix mobile serveur005.chm", NumHelp(MoiMême))  
// Ouverture de la page d'aide associée à la fenêtre en cours  
wHelp( "Aide winnix mobile serveur005.chm", NumHelp(MoiMême))
```

## FEN\_serveur

### Procédures

#### Procédure locale actualiseMDP

```
// Résumé : Permet de prendre en compte dynamiquement les changements de mot de passe dans le fichier Config.txt
// Syntaxe :
// actualiseMDP ()
//
// Paramètres :
//   Aucun
// Valeur de retour :
//   Aucune
//
// Exemple :
// ThreadExécute("actualiseMDP",threadNormal,actualiseMDP)
//
PROCEDURE actualiseMDP()
BOUCLE
  nIndexFichier est un entier = fOuvre("Config.txt",foLecture)
  SI nIndexFichier <> -1 ALORS
    gsMotDePasse = ExtraireChaîne(fLitLigne(nIndexFichier),2,":")
  FIN
  ThreadPause(100)
FIN
```

#### Procédure locale ajouteProspect

```
// Résumé : Permet d'ajouter un nouveau prospect dans la base, pour cela, la fonction reçoit par socket le Nom du nouveau Client et le nom de l'interlocuteur
//
// Syntaxe :
// ajouteProspect (<sThreadKiller> est chaîne UNICODE)
//
// Paramètres :
//   sThreadKiller (chaîne UNICODE) : contient le nom du Thread qui a lancé la fonction, qui est aussi l'adresse à laquelle envoyer une réponse (dans le socket)
//
// Valeur de retour :
//   Aucune
//
```

```
// Exemple :
// Indiquez ici un exemple d'utilisation.
//
PROCEDURE ajouteProspect(sThreadKiller est une chaîne UNICODE)
    nouveauClient est une Source de Données
    sInfosNouveauClient est une chaîne ANSI = UnicodeVersAnsi(SocketLit(sThreadKiller,Vrai))
    sNomSociete est une chaîne ANSI = ExtraireChaîne(sInfosNouveauClient,1,"#")
    sNomInterlocuteur est une chaîne ANSI = ExtraireChaîne(sInfosNouveauClient,2,"#")
    // # est le séparateur de données choisi pour échanger les informations
    requeteOk est un booléen = HExécuteRequêteSQL(nouveauClient,hRequêteDéfaut,"insert into Client (NomSociete,NomInterlocuteur) values (" +
    sNomSociete + "," + sNomInterlocuteur + ")")
    messOK est une chaîne UNICODE = "OK"

    SI requeteOk ALORS //on envoi un message à l'application cliente pour dire que l'ajout est effectif
        SocketEcrit(sThreadKiller,messOK)
    SINON
        SocketEcrit(sThreadKiller,"failure")
    FIN
    SocketFerme(sThreadKiller)

    ThreadArrête(sThreadKiller)
```

#### Procédure locale ajouteVisite

```
// Résumé : Cette fonction crée une nouvelle visite dans la base, pour cela elle reçoit par socket des données séparées par un #, exécute
// la requête REQ_Ajout_Visite

// Syntaxe :
//ajouteVisite (<threadkiller> est chaîne UNICODE)
//
// Paramètres :
//
threadkiller contient le nom du Thread qui a lancé la fonction, qui est aussi l'adresse à laquelle envoyer une réponse (dans le socket)
// valeur de retour :
// Aucune
//
// Exemple :
// ThreadExécute(sAdrClient,threadCopieComplèteContexteHFSQL,ajouteVisite,sAdrClient)
//
PROCEDURE ajouteVisite(threadkiller est une chaîne UNICODE)
    sdIDClient est une Source de Données
    sDonnéesAAjouter est une chaîne UNICODE = SocketLit(threadkiller,Vrai)
    HExécuteRequêteSQL(sdIDClient,
```



```

hRequêteDéfaut, "select IDClient from Client where NomSociete = " + ExtraitChaîne(sDonnéesAAjouter,3,"#") + """)
// on récupère l'IDClient à l'aide du NomSociete associé
HLitPremier(sdIDClient)
HLitDernier(Visite,IDVisite)
nNouvelIDVisite est un entier = HNbEnr(Visite) + 1
//la clé primaire de Visite est composée come suis : codeReprésentant*date*nombre d'enregistrements + 1

bResRequete est un booléen = HExécuteRequête(REQ_Ajout_Visite,hRequêteDéfaut,ExtraitChaîne(sDonnéesAAjouter,1,"#") + "" +
nNouvelIDVisite,ExtraitChaîne(sDonnéesAAjouter,2,"#"),sdIDClient.IDClient,ExtraitChaîne(sDonnéesAAjouter,4,"#"),ExtraitChaîne(
sDonnéesAAjouter,5,"#"),ExtraitChaîne(sDonnéesAAjouter,6,"#"),ExtraitChaîne(sDonnéesAAjouter,7,"#"),ExtraitChaîne(sDonnéesAAjouter,8,"#")
,ExtraitChaîne(sDonnéesAAjouter,9,"#"),ExtraitChaîne(sDonnéesAAjouter,10,"#"),ExtraitChaîne(sDonnéesAAjouter,11,"#"),ExtraitChaîne(
sDonnéesAAjouter,12,"#"),ExtraitChaîne(sDonnéesAAjouter,13,"#"),ExtraitChaîne(sDonnéesAAjouter,14,"#"),ExtraitChaîne(sDonnéesAAjouter,15
,"#"),ExtraitChaîne(sDonnéesAAjouter,16,"#"),ExtraitChaîne(sDonnéesAAjouter,17,"#"),ExtraitChaîne(sDonnéesAAjouter,18,"#"),ExtraitChaîne(
sDonnéesAAjouter,19,"#"),ExtraitChaîne(sDonnéesAAjouter,20,"#"),
ExtraitChaîne(sDonnéesAAjouter,21,"#"),ExtraitChaîne(sDonnéesAAjouter,22,"#"),ExtraitChaîne(sDonnéesAAjouter,23,"#"),ExtraitChaîne(
sDonnéesAAjouter,24,"#"),ExtraitChaîne(sDonnéesAAjouter,25,"#"),ExtraitChaîne(sDonnéesAAjouter,26,"#"),ExtraitChaîne(sDonnéesAAjouter,27,"#")
), ExtraitChaîne(sDonnéesAAjouter,28,"#"))

sMessOk est une chaîne UNICODE = "OK"

SI bResRequete ALORS
    SocketEcrit(threadkiller,sMessOk)
SINON
    SocketEcrit(threadkiller,"failure")
FIN
ThreadArrête(threadkiller)

```

### Procédure locale Connexion

```

// Résumé : cherche les clients voulant se connecter, et renvoi "connexion" dans le socket si le codeCommercial passé côté client est
correcte, sinon ne renvoi rien

// Syntaxe :
//Connexion ()
//
// Paramètres :
// Aucun
// Valeur de retour :
// Aucune
//
// Exemple :
// ThreadExecute("attenteConnexion", threadnormal, Connexion())
//
// threadkiller (chaîne UNICODE) : <indiquez ici le rôle de threadkiller>
PROCEDURE Connexion()
BOUCLE
    SI SocketAttendConnexion("Serveur",1000) ALORS

```

```

sAdrClient est une chaîne UNICODE = SocketAccepte("Serveur") //cette adresse est unique pour chaque client qui se connecte
sMessage est une chaîne UNICODE = SocketLit(sAdrClient,Faux)
ecouteActions(sAdrClient,sMessage)
FIN

```

---

**Procédure locale ecouteActions**


---

```

// Résumé : Ecoute les instructions de l'application cliente sous forme de mot clé, puis execute la fonction associée dans un thread
// Syntaxe :
//ecouteActions (<sAdrClient> est chaîne UNICODE, <sInstruction> est chaîne UNICODE)
//
// Paramètres :
// sAdrClient (chaîne UNICODE) : Adresse de l'application cliente récupérée via SocketAccept()
// sInstruction (chaîne UNICODE) : mot-clé permettant à la fonction de savoir quelle fonction lancer
// valeur de retour :
// Aucune
//
// Exemple :
// ThreadExecute("threadEcoute", threadnormal, ecouteActions)
//
PROCEDURE ecouteActions(sAdrClient est une chaîne UNICODE, sInstruction est une chaîne UNICODE)
    SELON sInstruction
    CAS "nomClient"
        ThreadExécute(sAdrClient,threadCopieComplèteContexteHFSQL,retourNomClient,sAdrClient)
    CAS "ajouteProspect"
        ThreadExécute(sAdrClient,threadCopieComplèteContexteHFSQL,ajouteProspect,sAdrClient)
    CAS "connexion"
        SIDRepresentantEtMDP est une chaîne UNICODE = SocketLit(sAdrClient,Vrai)
        SIDRepresentant est une chaîne UNICODE = ExtraitChaîne(SIDRepresentantEtMDP,1,"#")
        sMotDePasse est une chaîne UNICODE = ExtraitChaîne(SIDRepresentantEtMDP,2,"#")

        HLitPremier(Commercial)
        TANTQUE HEnDehors(Commercial) = Faux
            SI Commercial.CodeCommercial = SIDRepresentant _ET_ sMotDePasse = gsMotDePasse ALORS
                SocketEcrit(sAdrClient,gsConnexion)
            FIN
        HLitSuivant(Commercial)
    FIN
    SocketFerme(sAdrClient)
    CAS "ajouter"
        ThreadExécute(sAdrClient,threadCopieComplèteContexteHFSQL,ajoutevisite,sAdrClient)
    CAS "visualisation"
        ThreadExécute(sAdrClient,threadCopieComplèteContexteHFSQL,visualisation,sAdrClient)
    CAS "modifier"
        ThreadExécute(sAdrClient,threadCopieComplèteContexteHFSQL,modifiervisite,sAdrClient)

```

AUTRE CAS  
FIN

```
//ThreadEnvoieSignal(sAdrClient)
```

---

### Procédure locale modifierVisite

---

```
// Résumé : Modifie une visite en y insérant les données reçues dans le socket (cf ajoutevisite), execute la requête REQ_Modif_Visite
// Syntaxe :
//modifiervisite (<threadkiller> est chaîne UNICODE)
//
// Paramètres :
//
threadkiller (chaîne UNICODE) : contient le nom du Thread qui a lancé la fonction, qui est aussi l'adresse à laquelle envoyer une réponse
(dans le socket)

// valeur de retour :
// Aucune
//
// Exemple :
// ThreadExécute(sAdrClient,threadCopieComplèteContexteHFSQL,modifiervisite,sAdrClient)
//
PROCEDURE modifiervisite(threadkiller est une chaîne UNICODE)

sDonnéesModifiées est une chaîne UNICODE = SocketLit(threadkiller,Vrai)
SI sDonnéesModifiées <> "" ALORS
    sdIDclient est une Source de Données

    HExécuteRequêteSQL(sdIDclient,hRequêteDéfaut,"select IDClient from Client where NomSociete = " + ExtraitChaîne(sDonnéesModifiées,3,"#") + "")

    bRequeteOK est un booléen = HExécuteRequête(REQ_Modif_Visite,hRequêteDéfaut,ExtraitChaîne(sDonnéesModifiées,1,"#"),ExtraitChaîne(
sDonnéesModifiées,2,"#"),sdIDclient.IDClient,ExtraitChaîne(sDonnéesModifiées,4,"#"),ExtraitChaîne(sDonnéesModifiées,5,"#"),ExtraitChaîne(
sDonnéesModifiées,6,"#"),ExtraitChaîne(sDonnéesModifiées,7,"#"),ExtraitChaîne(sDonnéesModifiées,8,"#"),ExtraitChaîne(sDonnéesModifiées,
9,"#"),ExtraitChaîne(sDonnéesModifiées,10,"#"),ExtraitChaîne(sDonnéesModifiées,11,"#"),ExtraitChaîne(sDonnéesModifiées,12,"#"),
ExtraitChaîne(sDonnéesModifiées,13,"#"),ExtraitChaîne(sDonnéesModifiées,14,"#"),ExtraitChaîne(sDonnéesModifiées,15,"#"),ExtraitChaîne(
sDonnéesModifiées,16,"#"),ExtraitChaîne(sDonnéesModifiées,17,"#"),ExtraitChaîne(sDonnéesModifiées,18,"#"),ExtraitChaîne(
sDonnéesModifiées,19,"#"),ExtraitChaîne(sDonnéesModifiées,20,"#"),
ExtraitChaîne(sDonnéesModifiées,21,"#"),ExtraitChaîne(sDonnéesModifiées,22,"#"),ExtraitChaîne(sDonnéesModifiées,23,"#"),ExtraitChaîne(
sDonnéesModifiées,24,"#"),ExtraitChaîne(sDonnéesModifiées,25,"#"),ExtraitChaîne(sDonnéesModifiées,26,"#"),ExtraitChaîne(
sDonnéesModifiées,27,"#"), ExtraitChaîne(sDonnéesModifiées,28,"#"))

    sMessOk est une chaîne UNICODE = "OK"

    SI bRequeteOK ALORS
        SocketEcrit(threadkiller,sMessOk)
    SINON
        SocketEcrit(threadkiller,
```

```

"failure")
  FIN

FIN
Info(sDonnéesModifiées)
SocketFerme(threadkiller)
ThreadArrête(threadkiller)

```

---

**Procédure locale retourNomClient**


---

```

// Résumé : retourne nom par nom tous les noms contenant la chaîne de caractère envoyées par le client, pour cela on fait une requete sur
la table Client

// Syntaxe :
// retourNomClient (<sThreadkiller> est chaîne UNICODE)
//
// Paramètres :
// sThreadkiller (chaîne UNICODE) : nom du thread qui a lancé la fonction, sert à arrêter le thread en fin de traitement
// Valeur de retour :
// Aucune
//
// Exemple :
// CAS "ajout"
// ThreadExecute(<adresseClient>, threadnormal, retourNomClient, <adresseClient>)
//

```

```

PROCEDURE retourNomClient(sThreadkiller est une chaîne UNICODE)
  nomsClients est une Source de Données
  nomCherche est une chaîne UNICODE = SocketLit(sThreadkiller, Vrai)
  HExécuteRequêtesSQL(nomsClients, hRequêteDéfaut, "select NomSociete from Client where NomSociete like '%" + nomCherche + "%'")
  // la requete cherche le nom des Société dans le nom de Société contient ce qui est passé dans le socket

  HlitPremier(nomsClients)
  sNomEnvoye est une chaîne UNICODE = ""
  TANTQUE HEnDehors(nomsClients) = Faux
    sNomEnvoye += nomsClients.NomSociete + "#"
    HlitSuivant(nomsClients)
  FIN
  SocketEcrit(sThreadkiller, sNomEnvoye)
  SocketFerme(sThreadkiller)
  ThreadArrête(sThreadkiller)

```

---

**Procédure locale Visualisation**


---

```

// Résumé : execute la requete REQ_Visualisation qui prend comme paramètre <codeCommercial> et <nomClient>, et va écrire dans le socket
ligne par ligne le résultat de la requete

```

```
// Syntaxe :
//visualisation (<threadkiller> est chaîne UNICODE)
//
// Paramètres :
//
threadkiller (chaîne UNICODE) :contient le nom du Thread qui a lancé la fonction, qui est aussi l'adresse à laquelle envoyer une réponse
(dans le socket)

// valeur de retour :
// Aucune
//
// Exemple :
// CAS "visualiser"
// ThreadExecute(<adresseClient>, threadnormal, visualisation, <codeCommercial>, <nomClient>, <threadkiller>)
// AUTRES CAS
```

```
PROCEDURE visualisation(threadkiller est une chaîne UNICODE)
  sCommercialletClient est une chaîne UNICODE = SocketLit(threadkiller,Faux,1000)
  codeCommercial est une chaîne = ExtraitChaîne(sCommercialletClient,1,"#")
  nomClient est une chaîne = ExtraitChaîne(sCommercialletClient,2,"#")

  sdIDClient est une Source de Données

  HExécuteRequêtesSQL(sdIDClient,hRequêteDéfaut,"select IDClient from Client where NomSociete = " + nomClient + "")

  HLitPremier(sdIDClient)
  HExécuteRequête(REQ_Visualisation,hRequêteDéfaut,codeCommercial,sdIDClient.IDClient)
  infosAPasser est une chaîne UNICODE = ""
  nBrVisites est un entier = 0
  HLitPremier(REQ_Visualisation)
  TANTQUE HEnDehors(REQ_Visualisation) = Faux

    infosAPasser += REQ_Visualisation.IDVisite + "#" + nomClient + "#" + REQ_Visualisation.CommentaireActivite + "#" +
    REQ_Visualisation.Collaborateur + "#" + REQ_Visualisation.Roulement +
    "#" + REQ_Visualisation.Guidage + "#" + REQ_Visualisation.Transmission + "#" + REQ_Visualisation.Etancheite + "#" +
    REQ_Visualisation.Graissecolle + "#" +
    REQ_Visualisation.NouveauxProduits + "#" + REQ_Visualisation.PresentationRS + "#" + REQ_Visualisation.Decouverte + "#" +
    REQ_Visualisation.Negociation + "#" + REQ_Visualisation.Commande +
    "#" + REQ_Visualisation.Relationel + "#" + REQ_Visualisation.SAV + "#" + REQ_Visualisation.Pneumatique + "#" +
    REQ_Visualisation.Achat + "#" + REQ_Visualisation.BE + "#" +
    REQ_Visualisation.Entretien + "#" + REQ_Visualisation.CodeDuree + "#" + REQ_Visualisation.Commentaire + "#" +
    REQ_Visualisation.IDTypeVisite + "#" + REQ_Visualisation.clientProspect + "#" + REQ_Visualisation.Presence + "#" +
    REQ_Visualisation.date + "{" // le # est le séparateur de champs choisi pour distinguer les données
    //Info(infosAPasser)
    nBrVisites++
    HLitSuivant(REQ_Visualisation)
  FIN
  infosAPasser += "}" + nBrVisites
  SI infosAPasser <> "#0" ALORS
```

```
        SocketEcrit(threadkiller, infosAPasser)
FIN
SocketFerme(threadkiller)
ThreadArrête(threadkiller,10)
```

## Partie 5

**Requête**

## REQ\_Visualisation

Image

Afficher tous les ID\_Visite, CodeCommercial, IDClient, DTypeVisite, IDVendeur, date, dateNotification, nomTignesSales, clientProspect, Achat, BE, Entretien, Roulement, Guidage, Transmission, Blanche te, Gris lisee co le, Nouveaux Produits, PresentationRS, Decouverte, Neoposition, Commande, Relationel, SAV, Pneumatique, Commentaire Activite, CodeDuree, Commentaire, Collaborateur et Presence te le que CodeCommercial est egal a ParamCode Commercial ET IDClient est egal a ParamIDClient.





## REQ\_Visualisation

### Informations générales

REQ_Visualisation	
Nom logique de la requête	REQ_Visualisation
Nom physique de la requête	C:\Mes Projets\winnix mobile serveur\REQ_Visualisation.WDR
Libellé de la requête	
Type de requête	Requête de sélection
Analyse associée	C:\Mes Projets\winnix mobile serveur\winnix mobile serveur.ana\winnix mobile serveur.wda

## REQ\_Visualisation

### Informations sur le résultat

Rubriques résultat	Origine	Type	Sim	Cal	Agr	Tri
IDVisite	Visite.IDVisite	Chaîne	○			
CodeCommercial	Visite.CodeCommercial	Chaîne	○			
IDClient	Visite.IDClient	Entier sur 4 octets	○			
IDTypeVisite	Visite.IDTypeVisite	Entier sur 8 octets	○			
IDMarque	Visite.IDMarque	Entier sur 8 octets	○			
date	Visite.date	Date (aaaammjj)	○			
dateModification	Visite.dateModification	Date (aaaammjj)	○			
nbrLignesSaisies	Visite.nbrLignesSaisies	Entier sur 4 octets	○			
clientProspect	Visite.clientProspect	Caractère	○			
Achat	Visite.Achat	Booléen	○			
BE	Visite.BE	Booléen	○			
Entretien	Visite.Entretien	Booléen	○			
Roulement	Visite.Roulement	Booléen	○			

Rubriques résultat	Origine	Type	Sim	Cal	Agr	Tri
Guidage	Visite.Guidage	Booléen	<input type="radio"/>			
Transmission	Visite.Transmission	Booléen	<input type="radio"/>			
Etancheite	Visite.Etancheite	Booléen	<input type="radio"/>			
Graissecolle	Visite.Graissecolle	Booléen	<input type="radio"/>			
NouveauxProduits	Visite.NouveauxProduits	Booléen	<input type="radio"/>			
PresentationRS	Visite.PresentationRS	Booléen	<input type="radio"/>			
Decouverte	Visite.Decouverte	Booléen	<input type="radio"/>			
Negociation	Visite.Negociation	Booléen	<input type="radio"/>			
Commande	Visite.Commande	Booléen	<input type="radio"/>			
Relationel	Visite.Relationel	Booléen	<input type="radio"/>			
SAV	Visite.SAV	Booléen	<input type="radio"/>			
Pneumatique	Visite.Pneumatique	Booléen	<input type="radio"/>			
CommentaireActivite	Visite.CommentaireActivite	Chaîne	<input type="radio"/>			
CodeDuree	Visite.CodeDuree	Entier sur 4 octets	<input type="radio"/>			
Commentaire	Visite.Commentaire	Chaîne	<input type="radio"/>			
Collaborateur	Visite.Collaborateur	Chaîne	<input type="radio"/>			
Presence	Visite.Presence	Entier sur 4 octets	<input type="radio"/>			

Simple\*: Rubrique simple  
Agréгат\*: Rubrique agrégat

Calculée\*: Rubrique calculée

Paramètres de la requête	Test*	Valeur de test
ParamCodeCommercial	<input type="radio"/>	
ParamIDClient	<input type="radio"/>	

Test\*: Ce paramètre est utilisé dans le test de la requête

Paramètres avancés	
Afficher les doublons	<Oui>
Mode d'affichage	Afficher tous les enregistrements
HSansCorrection en mode test	<Non>

## REQ\_Visualisation

### Informations complémentaires

Conditions de sélection	Paramètre utilisé
CodeCommercial est égal à 'ParamCodeCommercial'	ParamCodeCommercial
IDClient est égal à ParamIDClient	ParamIDClient

## REQ\_Visualisation

### Code

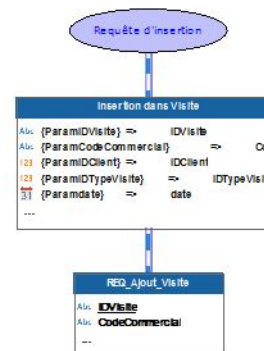
#### Code SQL de REQ\_Visualisation

```
SELECT
  Visite.IDVisite AS IDVisite,
  Visite.CodeCommercial AS CodeCommercial,
  Visite.IDClient AS IDClient,
  Visite.IDTypeVisite AS IDTypeVisite,
  Visite.IDMarque AS IDMarque,
  Visite.date AS DATE,
  Visite.dateModification AS dateModification,
  Visite.nbrLignesSaisies AS nbrLignesSaisies,
  Visite.clientProspect AS clientProspect,
  Visite.Achat AS Achat,
  Visite.BE AS BE,
  Visite.Entretien AS Entretien,
  Visite.Roulement AS Roulement,
  Visite.Guidage AS Guidage,
  Visite.Transmission AS Transmission,
  Visite.Etancheite AS Etancheite,
  Visite.Graissecolle AS Graisscolle,
```

```
Visite.NouveauxProduits AS NouveauxProduits,  
Visite.PresentationRS AS PresentationRS,  
Visite.Decouverte AS Decouverte,  
Visite.Negociation AS Negociation,  
Visite.Commande AS Commande,  
Visite.Relationel AS Relationel,  
Visite.SAV AS SAV,  
Visite.Pneumatique AS Pneumatique,  
Visite.CommentaireActivite AS CommentaireActivite,  
Visite.CodeDuree AS CodeDuree,  
Visite.Commentaire AS Commentaire,  
Visite.Collaborateur AS Collaborateur,  
Visite.Presence AS Presence  
FROM  
  Visite  
WHERE  
  Visite.CodeCommercial = {ParamCodeCommercial}  
  AND Visite.IDClient = {ParamIDClient}
```

## REQ\_Ajout\_Visite

Image



**REQ\_Ajout\_Visite****Informations générales**

REQ_Ajout_Visite	
Nom logique de la requête	REQ_Ajout_Visite
Nom physique de la requête	C:\Mes Projets\winnix mobile serveur\REQ_Ajout_Visite.WDR
Libellé de la requête	
Type de requête	Requête d'insertion
Analyse associée	C:\Mes Projets\winnix mobile serveur\winnix mobile serveur.ana\winnix mobile serveur.wda

**REQ\_Ajout\_Visite****Informations sur le résultat**

Simple*:	Rubrique simple	Calculée*:	Rubrique calculée
Agrégat*:	Rubrique agrégat		

Paramètres de la requête	Test*	Valeur de test
ParamIDVisite	<input type="radio"/>	
ParamCodeCommercial	<input type="radio"/>	
ParamIDClient	<input type="radio"/>	
ParamIDTypeVisite	<input type="radio"/>	
Paramdate	<input type="radio"/>	
ParamdateModification	<input type="radio"/>	
ParamclientProspect	<input type="radio"/>	
ParamAchat	<input type="radio"/>	
ParamBE	<input type="radio"/>	
ParamEntretien	<input type="radio"/>	

Paramètres de la requête	Test*	Valeur de test
ParamRoulement	<input type="radio"/>	
ParamGuidage	<input type="radio"/>	
ParamTransmission	<input type="radio"/>	
ParamEtancheite	<input type="radio"/>	
ParamGraissecolle	<input type="radio"/>	
ParamNouveauxProduits	<input type="radio"/>	
ParamPresentationRS	<input type="radio"/>	
ParamDecouverte	<input type="radio"/>	
ParamNegociation	<input type="radio"/>	
ParamCommande	<input type="radio"/>	
ParamRelationel	<input type="radio"/>	
ParamSAV	<input type="radio"/>	
ParamPneumatique	<input type="radio"/>	
ParamCommentaireActivite	<input type="radio"/>	
ParamCodeDuree	<input type="radio"/>	
ParamCommentaire	<input type="radio"/>	
ParamCollaborateur	<input type="radio"/>	
ParamPresence	<input type="radio"/>	

Test\*: Ce paramètre est utilisé dans le test de la requête

Paramètres avancés	
Afficher les doublons	<Oui>
Mode d'affichage	Afficher tous les enregistrements
HSansCorrection en mode test	<Non>

## REQ\_Ajout\_Visite

Code

Code SQL de REQ\_Ajout\_Visite

```
INSERT INTO Visite
(
    IDVisite,
    CodeCommercial,
    IDClient,
    IDTypeVisite,
    DATE,
    dateModification,
    clientProspect,
    Achat,
    BE,
    Entretien,
    Roulement,
    Guidage,
    Transmission,
    Etancheite,
    Graissecolle,
    NouveauxProduits,
    PresentationRS,
    Decouverte,
    Negociation,
    Commande,
    Relationel,
    SAV,
    Pneumatique,
    CommentaireActivite,
    CodeDuree,
    Commentaire,
    Collaborateur,
    Presence
)
VALUES
(
    {ParamIDVisite},
    {ParamCodeCommercial},
    {ParamIDClient},
    {ParamIDTypeVisite},
    {Paramdate},
    {ParamdateModification},
    {ParamclientProspect},
    {ParamAchat},
    {ParamBE},
    {ParamEntretien},
    {ParamRoulement},
    {ParamGuidage},
    {ParamTransmission},
    {ParamEtancheite},
    {ParamGraissecolle},
    {ParamNouveauxProduits},
```



```
{ParamPresentationRS},  
{ParamDecouverte},  
{ParamNegociation},  
{ParamCommande},  
{ParamRelationI},  
{ParamSAV},  
{ParamPneumatique},  
{ParamCommentaireActivite},  
{ParamCodeDuree},  
{ParamCommentaire},  
{ParamCollaborateur},  
{ParamPresence}  
)
```

## REQ\_Modif\_Visite

Image

Code SQL de REQ_Modif_Visite
UPDATE Visite SET IDVisite = (ParamIDVisite), CodeCommercial = (ParamCodeCommercial), DClient = (ParamDClient), DTypVisite = (ParamDTypVisite), date = (Paramdate), dateIndication = (ParamdateIndication)

**REQ\_Modif\_Visite****Informations générales**

REQ_Modif_Visite	
Nom logique de la requête	REQ_Modif_Visite
Nom physique de la requête	C:\Mes Projets\winnix mobile serveur\REQ_Modif_Visite.WDR
Libellé de la requête	
Type de requête	Requête de modification
Analyse associée	C:\Mes Projets\winnix mobile serveur\winnix mobile serveur.ana\winnix mobile serveur.wda

**REQ\_Modif\_Visite****Informations sur le résultat**

Simple*:	Rubrique simple	Calculée*:	Rubrique calculée
Agrégat*:	Rubrique agrégat		

Paramètres de la requête	Test*	Valeur de test
ParamIDVisite	<input type="radio"/>	
ParamCodeCommercial	<input type="radio"/>	
ParamIDClient	<input type="radio"/>	
ParamIDTypeVisite	<input type="radio"/>	
Paramdate	<input type="radio"/>	
ParamdateModification	<input type="radio"/>	
ParamclientProspect	<input type="radio"/>	
ParamAchat	<input type="radio"/>	
ParamBE	<input type="radio"/>	
ParamEntretien	<input type="radio"/>	

Paramètres de la requête	Test*	Valeur de test
ParamRoulement	<input type="radio"/>	
ParamGuidage	<input type="radio"/>	
ParamTransmission	<input type="radio"/>	
ParamEtancheite	<input type="radio"/>	
ParamGraissecolle	<input type="radio"/>	
ParamNouveauxProduits	<input type="radio"/>	
ParamPresentationRS	<input type="radio"/>	
ParamDecouverte	<input type="radio"/>	
ParamNegociation	<input type="radio"/>	
ParamCommande	<input type="radio"/>	
ParamRelationel	<input type="radio"/>	
ParamSAV	<input type="radio"/>	
ParamPneumatique	<input type="radio"/>	
ParamCommentaireActivite	<input type="radio"/>	
ParamCodeDuree	<input type="radio"/>	
ParamCommentaire	<input type="radio"/>	
ParamCollaborateur	<input type="radio"/>	
ParamPresence	<input type="radio"/>	

Test\*: Ce paramètre est utilisé dans le test de la requête

Paramètres avancés	
Afficher les doublons	<Oui>
Mode d'affichage	Afficher tous les enregistrements
HSansCorrection en mode test	<Non>

## REQ\_Modif\_Visite

Code

Code SQL de REQ\_Modif\_Visite

**UPDATE**

Visite

**SET**

```
IDVisite = {ParamIDVisite},
CodeCommercial = {ParamCodeCommercial},
IDClient = {ParamIDClient},
IDTypeVisite = {ParamIDTypevisite},
DATE = {Paramdate},
dateModification = {ParamdateModification},
clientProspect = {ParamclientProspect},
Achat = {ParamAchat},
BE = {ParamBE},
Entretien = {ParamEntretien},
Roulement = {ParamRoulement},
Guidage = {ParamGuidage},
Transmission = {ParamTransmission},
Etancheite = {ParamEtancheite},
Graissecolle = {ParamGraissecolle},
NouveauxProduits = {ParamNouveauxProduits},
PresentationRS = {ParamPresentationRS},
Decouverte = {ParamDecouverte},
Negociation = {ParamNegociation},
Commande = {ParamCommande},
Relationel = {ParamRelationel},
SAV = {ParamSAV},
Pneumatique = {ParamPneumatique},
CommentaireActivite = {ParamCommentaireActivite},
CodeDuree = {ParamCodeDuree},
Commentaire = {ParamCommentaire},
Collaborateur = {ParamCollaborateur},
Presence = {ParamPresence}
```

**WHERE**

Visite.IDVisite = {ParamIDVisite}

## Partie 6

### Table des matières

# Table des matières

## Projet winnix mobile serveur

### 3 En-tête Partie 1

#### 3 ○ En-tête

### 5 Projet Partie 2

- 5 ○ Informations générales
- 5 ○ Liste des éléments du projet
- 6 ○ Tableau de bord
- 6 ○ Code
- 6 ○ Erreurs, warnings et informations
- 6 ○ Statistiques sur le code

### 9 Analyse Partie 3

- 9 ○ Graphe
- 11 ○ Informations générales
- 11 ○ Dictionnaire des rubriques
- 13 ○ Liaisons
- 14 ○ Connexions
- 15 ○ Fichiers et rubriques
  - 15 ○ Client
  - 17 ○ Commercial
  - 18 ○ Marque
  - 19 ○ TypeVisite
  - 20 ○ Visite

### 23 Fenêtre WinDev Partie 4

- 23 ○ FEN\_serveur
- 23 ○ Image

- 24 ○ Informations générales
- 26 ○ Informations sur les champs
- 29 ○ Code
- 30 ○ Code des champs
- 31 ○ Procédures

### 40 Requête Partie 5

- 40 ○ REQ\_Visualisation
  - 40 ○ Image
  - 41 ○ Informations générales
  - 41 ○ Informations sur le résultat
  - 43 ○ Informations complémentaires
  - 43 ○ Code
- 45 ○ REQ\_Ajout\_Visite
  - 45 ○ Image
  - 46 ○ Informations générales
  - 46 ○ Informations sur le résultat
  - 47 ○ Code
- 50 ○ REQ\_Modif\_Visite
  - 50 ○ Image
  - 51 ○ Informations générales
  - 51 ○ Informations sur le résultat
  - 52 ○ Code