
	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		


Suivi des versions-révisions et des validations du document.			
<p>Ce document annule et remplace tout document diffusé de version-révision antérieure.</p> <p>Dès réception de ce document, les destinataires ont pour obligation de détruire les versions-révisions antérieures, toutes les copies, et de les remplacer par cette version.</p> <p>Si les versions-révisions antérieures sont conservées pour mémoire, les destinataires doivent s'assurer qu'elles ne peuvent être confondues avec cette présente version-révision dans leur usage courant.</p>			
Version.	Date.	Auteurs.	Création, modification ou validation.
A	20 oct. 2003.	JPD.	Création.

	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		

1 Tables


1.1 Table des matières

1	Tables	2
1.1	Table des matières	2
1.2	Table des illustrations	3
2	Références	4
2.1	Glossaire	4
2.2	Ressources	4
3	Introduction	5
3.1	Objet du document	5
3.2	Audience	5
3.3	Pré-requis	5
4	Interaction avec l'environnement	6
4.1	Description	6
4.2	Paramètres	6
4.2.1	Paramètre Profil	6
4.3	Particularités	6
4.3.1	Compilation	6
4.3.2	Exécution	7
4.4	Application Program Interfaces	7
4.4.1	_close – Win32	7
4.4.2	_fileno – Win32	7
4.4.3	_fstat – Win32	8
4.4.4	_open – Win32	8
4.4.5	_setmode – Win32	9
5	Choix techniques	10
5.1	Architecture	10
5.2	Persistance	10
5.3	Trace	11
6	Modèle de données	12
6.1	Entité Adresse	13
6.2	Entité Contact	14
6.3	Entité Société	15
7	Composants techniques	16

	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		

1.2 Table des illustrations

Tableau 1 – Paramètre Profil.....	6
Tableau 2 – Modèle physique de données du module <i>Contacts</i>	12
Tableau 3 – Entité Adresse	14
Tableau 4 – Entité Contact	15
Tableau 5 – Entité Société.....	15
Tableau 6 – Composants techniques du module	16

	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		


2 Références

2.1 Glossaire

Liste des définitions des termes employés.	
Ce tableau recense tous les termes, les concepts particuliers ainsi que les abréviations employés dans ce document.	
Terme, concept, abrégé.	Définition du terme, du concept ou de l'abréviation.

2.2 Ressources

Liste des documents applicables et en référence.		
Un document est applicable à partir du moment où son contenu est validé et que l'activité ou le projet fait partie de son périmètre d'application. Il est obligatoire d'appliquer son contenu.		
Un document est en référence à partir du moment où son contenu n'est pas validé ou que l'activité ou le projet ne fait partie de son périmètre d'application. Il est recommandé d'appliquer son contenu mais cela n'est pas obligatoire.		
Un document applicable est indicé par A1, A2, A3 , etc. Un document en référence est indicé par R1, R2, R3 , etc.		
Index.	Nom du document.	Commentaire.
A1	UpComp-Plan Qualité-000005	Méthode documentaire.
A2	UpComp-Plan Qualité-000006	Processus de management de projet.
A3	UpComp-Plan Qualité-000046	Méthode de spécification technique d'un module.
A4	UpComp-Contacts-000002	Plan documentaire du projet.
A5	UpComp-Contacts-000007	Spécification des besoins du système.
A6	UpComp-Contacts-000013	Spécification d'architecture du système.
A7	UpComp-Contacts-000018	Spécification technique d'interface.
A8	UpComp-Contacts-000015	Spécification technique des composants matériels.
A9	UpComp-Contacts-000018	Spécification technique d'un composant logiciel.

	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		

3 Introduction

3.1 Objet du document

L'objet de ce document est de décrire le contenu technique du module logiciel **Contacts** pour le **Contacts**.

Ce document est rédigé et approuvé par la **Maîtrise d'Oeuvre (MOE)**.

3.2 Audience

Ce document s'adresse aux :

- **Directeurs de projets et chefs de projets.**
Pour la compréhension du module technique.
- **Ingénieurs de développement.**
Pour savoir comment est conçu le module technique.


Pour aider ces personnes à remplir le document **Spécification technique d'un module**, leur manager et la cellule de support projet se tiennent à leur disposition.

3.3 Pré-requis

Le pré-requis est la connaissance des documents suivants :

- **Méthode documentaire** [A1].
- **Processus de management de projet** [A2].
- **Méthode de spécification technique d'un module** [A3].

Nous rappelons que tous les documents applicables ou référencés pour le projet **Contacts** sont tracés dans le **Plan documentaire** [A4].

	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		

4 Interaction avec l'environnement

4.1 Description

L'objet du module logiciel **Contacts** est de gérer les contacts de la société **Up! Company**.
Voici les différents types de contacts :

- Banquiers.
- Clients.
- Distributeurs.
- Employés.
- Fournisseurs.
- Investisseurs.
- Partenaires.

Il existe différents profils d'usage du module, la liste n'étant pas exhaustive :

- Administratifs.
- Directeurs.
- Employés.

4.2 Paramètres

4.2.1 Paramètre Profil

Le paramètre **Profil** permet de spécifier le profil d'usage du module Contacts. Il est lu dans la base de registres dans **HKEY_LOCAL_SOFTWARE\UpCompany\Contacts**.

Voici la liste des valeurs admissibles :

Valeur du paramètre.	Sémantique.
Administratif.	Pour un profil administratif.
Directeur.	Pour un profil directeur.
Employé.	Pour un profil employé.

Tableau 1 – Paramètre Profil

Voici un exemple d'emploi :


Profil=Administratif

4.3 Particularités

4.3.1 Compilation

Voici les particularités de compilation du module **Contacts** pour le projet **Contacts** :

- La définition **WIN32** spécifie de produire un exécutable pour **Windows**.
- La définition **LINUX20** spécifie de produire un exécutable pour **Linux**.

	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		

4.3.2 Exécution

Voici les particularités d'exécution du module **Contacts** pour le projet **Contacts** :

- La présence de la variable environnement **HTTP_HOME** spécifie que le module fonctionne en **Intranet**. Sinon, il fonctionne en client-serveur.

4.4 Application Program Interfaces

Trois sortes d'**Application Program Interfaces (API)** sont utilisées :

- Celles de **Posix**.
- Celles de **Unix**.
- Celles de **Windows**.

Elles sont toutes documentées sur le site **Internet** d'**Up ! Company** dans la section **Normes et standards**, sauf pour celles énumérées ci-après :

4.4.1 _close – Win32

4.4.1.1 Prototypes

```
int _close(int NumeroFichier);
```

4.4.1.2 Description

Ferme le fichier identifié par **NumeroFichier**.

Le résultat est inutile.

4.4.1.3 Exemple

```
int NumeroFichier;
```

```
NumeroFichier=_open((LPCSTR)"c:\\tmp\\essai.txt", _O_BINARY);
```

```
...
```

```
_close(NumeroFichier);
```

4.4.1.4 Avertissement

Néant.

4.4.1.5 Voir aussi

_open pour ouvrir le fichier.


4.4.2 _fileno – Win32

4.4.2.1 Prototypes

```
int _fileno(FILE *Fichier);
```

4.4.2.2 Description

Lit le numéro de fichier du descripteur de fichier **Fichier**.

	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		

Le résultat est le numéro de fichier.

4.4.2.3 Exemple

```
int NumeroFichier;
FILE *Fichier;
```

```
NumeroFichier=_fileno(Fichier);
```

4.4.2.4 Avertissement

Néant.

4.4.3 `_fstat` – Win32

4.4.3.1 Prototypes

```
int _fstat(int NumeroFichier, struct _stat *Informations);
```

4.4.3.2 Description

Lit les caractéristiques du fichier identifié par *NumeroFichier* dans le descripteur *Informations*.
Le résultat est -1 si le fichier n'existe pas.

4.4.3.3 Exemple

```
int NumeroFichier;
struct _stat Stat;

if (_fstat(Fichier,&Stat)==-1)
{
...
}
```

4.4.3.4 Avertissement

Néant.

4.4.3.5 Voir aussi

`_open` pour ouvrir un fichier.


4.4.4 `_open` – Win32

4.4.4.1 Prototypes

```
int _open(LPCSTR NomFichier, int Mode);
```

4.4.4.2 Description

Ouvre le fichier *NomFichier* dans le mode *Mode*.
Le résultat est le numéro de fichier.

	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		

4.4.4.3 Exemple

```
int NumeroFichier;
```

```
NumeroFichier=_open((LPCSTR)"c:\\tmp\\essai.txt", _O_BINARY);
```

```
...
```

```
_close(NumeroFichier);
```

4.4.4.4 Avertissement

Néant.

4.4.4.5 Voir aussi

`_close` pour fermer le fichier.

4.4.5 `_setmode` – Win32

4.4.5.1 Prototypes

```
int _setmode(int Fichier, int Mode);
```

4.4.5.2 Description

Change le mode d'ouverture du fichier *Fichier* avec le nouveau mode *Mode*.

Le résultat est inutile.

4.4.5.3 Exemple

```
int NumeroFichier;
```


```
_setmode(NumeroFichier, _O_BINARY);
```

4.4.5.4 Avertissement

Néant.

4.4.5.5 Voir aussi

`_fileno` pour obtenir le numéro de fichier d'un descripteur de fichier.

	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		

5 Choix techniques

5.1 Architecture

L'objet est l'architecture de **Contacts** en terme d'interaction avec le système d'exploitation

Les différentes alternatives possibles sont les suivantes :

- Utiliser **Up ! Virtual Technical Machine**.
- Utiliser le langage **C**.
- Utiliser le langage **Perl**.
- Utiliser le langage **Java**.
- Utiliser le langage **Php**.

Le choix retenu est **C** pour la raison suivante :

- **Afin d'être léger à la diffusion.**
La machine de production est actuellement exploitée via un hébergeur. Nous ne pouvons pas administrer la machine en direct.
- **Afin d'être migré pour Up ! Virtual Technical Machine dans un premier temps.**
Le découpage logique du module isole les **APIs** système à la manière **Up ! System**.
- **Afin d'être migré en Up ! 5GL dans un second temps.**
Cela nécessite, d'une part, l'hébergement du site en interne, et, d'autre part, la mise en oeuvre des pilotes **Http** et **Sntp**.

En cas de changement de choix, les adaptations à apporter sont les suivantes :

- Pour la première migration, changement des **APIs** système et passage en **Unicode**.
- Pour la seconde migration, récupération du fonctionnel et ré encodage.

Le préfixe du module servant pour les symboles exportés est **UpsCom**.

5.2 Persistance


L'objet est la persistance de **Contacts** en terme de conservations des informations relatives aux adresses, aux contacts et aux sociétés.

Les différentes alternatives possibles sont les suivantes :

- Utiliser une base de données relationnelle.
- Utiliser des fichiers d'enregistrements.
- Utiliser des fichiers à section.

Le choix retenu est la base de données relationnelle **My Sql** pour la raison suivante :

- **Disponibilité.**
Notre hébergeur nous propose une instance d'une base de données de ce type d'une taille suffisante.
- **Gratuité.**
Il n'y a pas de licence à payer et le coût d'exploitation est déjà compris dans le contrat d'hébergement.

	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		

En cas de changement de choix, les adaptations à apporter sont les suivantes :


- Pour changer de base de données relationnelle, il faut reprendre le code de lecture, d'écriture et de modification qui utilisent les **APIs** de **My Sql**.
- Pour les fichiers d'enregistrement, cela n'est d'aucune utilité, compte tenu de la puissance des bases de données relationnelles.

5.3 Trace

Tous les appels de **Contacts** sont tracés au moins en début d'appel via un traitement du type :

```
if (ContactsFichierTrace)
{
    fprintf(ContactsFichierTrace, "Demarrer()\n");
    fflush(ContactsFichierTrace);
}
```

Le fichier **\$HOME/contacts/trace.txt** est alors produit si le paramètre de trace est activé.

	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		

6 Modèle de données

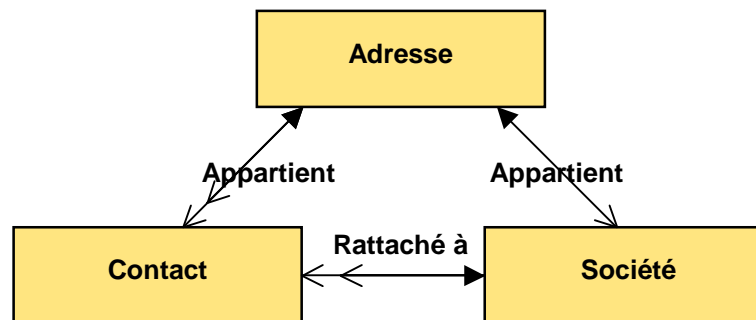



Tableau 2 – Modèle physique de données du module *Contacts*

	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		


Toutes les entités sont décrites en détail dans le fichier **contacts.e**. Voici leur description :

- **Adresse.**
Description d'une adresse d'un contact ou d'une société. Elle est conservée dans la table **Adresse**.
- **Contact.**
Description d'un contact d'**Up ! Company**. Elle est conservée dans la table **Contact**.
- **Société.**
Description d'une société en relation avec **Up ! Company**. Elle est conservée dans la table **Societe**.

6.1 Entité Adresse

Voici la description de l'entité **Adresse** correspondant à la table du même nom.

Propriétés.		
Dénomination.	Type.	Valeurs possibles.
Description.		
Batiment.	Varchar(255).	« Bâtiment A ».
Complément de l'adresse, tel le nom du bâtiment, le numéro de l'escalier, etc.		
CodePostal.	Number(7).	75001.
Numéro du code postal.		
Identifiant.	Number(7).	1.
Clé unique d'identification de l'adresse.		
IdentifiantContact.	Nul Ou Number(7).	1.
Clé unique d'identification du contact si l'adresse est celle d'un contact.		
IdentifiantSociete.	Nul Ou Number(7).	1.
Clé unique d'identification de la société si l'adresse est celle d'une société.		
Numero.	Number(7,2).	1.
Numéro de la rue.		
Rue.	Varchar(255).	« Boulevard Magenta ».
Nom de la rue, de l'avenue ou du boulevard.		
Ville.	Varchar(255).	« Paris ».
Nom de la ville.		
Méthodes.		
Dénomination et paramètres.		
Description.		
Creer().		
Création d'une nouvelle adresse.		

	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		


Lire(Critere : Varchar(255)).
Lecture d'un ou de plusieurs adresses en fonction du critère <i>Critere</i> .
Modifier().
Modification d'une adresse existante.
Supprimer().
Suppression d'une adresse existante.

Tableau 3 – Entité Adresse

6.2 Entité Contact

Voici la description de l'entité *Contact* correspondant à la table du même nom.

Propriétés.		
Dénomination.	Type.	Valeurs possibles.
Description.		
EEmail.	Varchar(255).	« j-duval@em-lyon.com ».
Adresse e-mail du contact.		
Identifiant.	Number(7).	1.
Clé unique d'identification du contact.		
IdentifiantAdresse.	Number(7).	1.
Clé unique d'identification de l'adresse du contact.		
IdentifiantSociete.	Number(7).	1.
Clé unique d'identification de la société du contact.		
Nom.	Varchar(255).	« DUVAL ».
Nom du contact.		
Prenom.	Varchar(255).	« Jean-Pierre ».
Prénom du contact.		
Telephone.	Number(10)	1.
Numéro de téléphone du contact.		
Méthodes.		
Dénomination et paramètres.		
Description.		
Creer().		
Création d'un nouveau contact.		
Lire(Critere : Varchar(255)).		
Lecture d'un ou de plusieurs contacts en fonction du critère <i>Critere</i> .		
Modifier().		

	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		

Modification d'un contact existant.

Supprimer().

Suppression d'un contact existant.


Tableau 4 – Entité Contact

6.3 Entité Société

Voici la description de l'entité **Société** correspondant à la table du même nom.

Propriétés.		
Dénomination.	Type.	Valeurs possibles.
Description.		
Identifiant.	Number(7).	1.
Clé unique d'identification de la société.		
IdentifiantAdresse.	Number(7).	1.
Clé unique d'identification de l'adresse de la société.		
RaisonSociale.	Varchar(255).	« Up ! Company ».
Raison sociale de la société.		
Siren.	Number(20).	« 0123456789 ».
Numéro d'immatriculation de la société.		
Méthodes.		
Dénomination et paramètres.		
Description.		
Creer().		
Création d'une nouvelle société.		
Lire(Critere : Varchar(255)).		
Lecture d'une ou de plusieurs sociétés en fonction du critère Critere .		
Modifier().		
Modification d'une société existante.		
Supprimer().		
Suppression d'une société existante.		

Tableau 5 – Entité Société

	Spécification technique d'un module	Date rédaction : 16 décembre 2003.
	Diffusion publique	Date validation :
Référence : UpComp-Meilleure pratique-000017-A Spécification technique d'un module.doc		

7 Composants techniques

Le module **Contacts** pour le projet **Contacts** est constitué des composants suivants :

Fichiers du module.	
<ul style="list-style-type: none"> Fichier contacts.h – Importation des définitions. Fichier contacts.e – Définitions partagées par les composants dont le modèle de données. 	
Composants.	Contacts.
Description.	
Gestion des contacts : <ul style="list-style-type: none"> Ajout, modification, recherche et suppression d'une adresse. Ajout, modification, recherche et suppression d'un contact. Ajout, modification, recherche et suppression d'une société. 	
Fichiers.	
<ul style="list-style-type: none"> Fichier contact.cpp – Code du composant. Fichier contact.h – Importation des définitions. Fichier contact.e – Définitions partagées par les composants dont le modèle de données. 	
Composants.	Demandes d'information.
Description.	
Gestion des demandes d'information : <ul style="list-style-type: none"> Ajout, modification, recherche et suppression d'une demande d'information. Réponse à une demande. Validation ou invalidation d'une demande. 	
Fichiers.	
<ul style="list-style-type: none"> Fichier demande.cpp – Code du composant. Fichier demande.h – Importation des définitions. Fichier demande.e – Définitions partagées par les composants dont le modèle de données. 	

Tableau 6 – Composants techniques du module

Fin de document