

| | | |
|--|--|---|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.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 | 23 nov. 2003. | JPD. | Création. |
| B | 10 juin 2003. | JPD. | Ajout de l'interface avec la console et de Invoquer(). |

| | | |
|--|--|--|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.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 |
| 4.3 | Particularités..... | 6 |
| 4.3.1 | Compilation..... | 6 |
| 4.3.2 | Exécution..... | 6 |
| 4.4 | Application Program Interfaces..... | 6 |
| 5 | Choix techniques..... | 7 |
| 5.1 | Gestion options..... | 7 |
| 5.1.1 | Extension d'un module..... | 7 |
| 5.1.2 | Extension d'un objet..... | 8 |
| 5.2 | Gestion de la mémoire..... | 8 |
| 5.2.1 | Allocateurs de mémoire..... | 8 |
| 5.2.2 | Table d'objets..... | 8 |
| 5.3 | Gestion des flux..... | 9 |
| 5.3.1 | Principe..... | 9 |
| 5.3.2 | Propriétés d'un flux..... | 10 |
| 5.3.3 | Méthodes d'un flux..... | 10 |
| 5.3.4 | Mode d'ouverture d'un flux..... | 11 |
| 5.3.5 | Exemple d'emploi des flux..... | 12 |
| 5.4 | Gestion des systèmes de fichiers..... | 12 |
| 5.4.1 | Organisation d'un système de fichiers..... | 13 |
| 5.4.2 | Méthodes d'un système de fichiers..... | 13 |
| 5.5 | Gestion de la console..... | 14 |
| 5.5.1 | Interface de la console..... | 15 |
| 5.5.2 | Gestionnaire de la console..... | 15 |
| 5.6 | Interface avec la machine..... | 16 |
| 5.6.1 | Section critique..... | 19 |
| 5.6.2 | Services pour Up ! Object Management System..... | 19 |
| 6 | Modèle de données..... | 21 |
| 7 | Composants techniques..... | 23 |

| | | |
|--|--|---|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

1.2 Table des illustrations

| | |
|---|----|
| Tableau 1 – Filtre des API AnalyserChaine et FormaterChaine | 8 |
| Tableau 2 – Filtre des API AnalyserChaine et FormaterChaine | 11 |
| Tableau 3 – Codification des masques d'ouverture de flux..... | 12 |
| Texte 4 – Exemple d'emploi des flux..... | 12 |
| Tableau 5 – Section critique d'Up ! Virtual Technical Machine | 19 |
| Diagramme 6 – Modèle physique des données publiques du module <i>Up ! System</i> | 21 |
| Tableau 7 – Glossaire du modèle physique des données publiques du module <i>Up ! System</i> | 22 |
| Tableau 8 – Composants techniques du module | 28 |

| | | |
|--|--|--|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.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-Upssys-000002 | Plan documentaire du projet. |
| A5 | UpComp-UpsVm-000003 | Plan de programmation. |
| A6 | UpComp-UpsVm-000004 | Programmation en C-- . |
| R7 | http://www.up-comp.com | Site Internet d'Up ! Application System . |
| A8 | UpComp-UpsOms-000003 | Spécification technique du module UpsOms . |
| A9 | UpComp-UpsKrn-000003 | Spécification technique du module UpsKrn . |

| | | |
|--|--|--|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

3 Introduction

3.1 Objet du document

L'objet de ce document est de décrire le contenu technique du module logiciel **Up ! System** pour le projet **Up ! Application System**.

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 composant logiciel** [A3].

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

| | | |
|--|--|--|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

4 Interaction avec l'environnement

4.1 Description

L'objet du module logiciel *Up ! System* est de

- Fournir les services d'exploitation du système de fichiers.
Il s'agit du système de fichiers natifs du système d'exploitation d'exécution.
- Fournir les services d'exploitation du système.
Pour les processus, les threads, les synchronisations, etc.
- Fournir des services de communication.
Ils sont indépendants du moyen de communication retenue.
- Mettre en oeuvre :
 - Les interfaces.
Il y a *Flux*.
 - Les types.
Il y a *Calendrier*, *Fichier*, *FichierListe*, *FileDAttente*, *PeriodeCalendrier*, *ProcessusListe*, *Segment*, *Synchronisation*, *Tache*, *Travail* et *UniteListee*.

4.2 Paramètres

Tous les paramètres sont documentés sur le [Site Internet d'Up ! System](#) [R7].

4.3 Particularités

4.3.1 Compilation

La définition `__CYGWIN__` identifie la plate-forme *Linux* émulée sous *Windows*.

4.3.2 Exécution

Néant.

4.4 Application Program Interfaces

Toutes les *Application Program Interfaces* (API) utilisées par *Up ! System* sont documentées sur le site [Internet](#) des ressources <http://www.up-comp.com> [R7].

| | | |
|--|--|--|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

5 Choix techniques

5.1 Gestion options

5.1.1 Extension d'un module

La gestion des adaptateurs et des pilotes est automatisée au travers du concept d'**option**. Ainsi, un module peut être étendu par un ou plusieurs autres modules selon les deux sémantiques suivantes :

- **Sous forme d'un type hérité.**
L'option est alors toujours active.
- **Sous forme d'une interface héritée.**
L'option est alors active suivant le contexte d'usage.
Voici les **Application Program Interfaces (API)** pour gérer les options :
- **UpsSys.ChargerOptions.**
Charge les modules qui sont les extensions d'un module particulier.
Optionnellement, le module étendu peut considérer ses extensions comme des interfaces s'il demande à allouer une table d'options. Ainsi, les options peuvent être activées au cas par cas, chaque session étant indépendante, via l'appel à l'**Application Program Interfaces (API) EnteteOption.ActiverOption** – Voir ci-après.
- **UpsSys.ChangerInstanceOption.**
Quand une extension d'un module change d'interface de données, suite à une commutation d'instance, il doit informer le module étendu de cela, via une **API** propriétaire, qui appelle **ChangerInstanceOption**.
- **UpsSys.ReferencerExtension.**
Une extension d'un module doit se référencer auprès du module étendu via cette **API**.
- **UpsSys.DechargerOptions.**
Décharge les modules qui sont les extensions d'un module.

5.1.1.1 En-tête d'une option

L'interface de traitements d'un module géré comme une option comporte obligatoirement les **API** suivantes regroupées dans **TypUpsVmTraitementsOption** :

- **UpsVtm.ActiverOption.**
Retourne **Vrai** si l'option doit être activée, au regard du contexte d'exécution défini, notamment par les paramètres transmis.
- **UpsVtm.LireLibelleOption.**
Lit le libellé d'une option pour l'afficher en tant qu'information.
- **UpsVtm.RechercherOption.**
Recherche une option donnée par son nom pour retrouver son interface de données et son interface de traitements.

A ce jour, les contextes suivants sont gérés via l'énuméré **EnuUpsVmCibleExecution** :

| Identifiant. | Description. |
|--------------|-----------------------|
| CE_Compiler. | <i>Up ! Compiler.</i> |
| CE_Engine. | <i>Up ! Engine.</i> |

| | | |
|--|--|--|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

Tableau 1 – Filtre des API AnalyserChaine et FormaterChaine

5.1.2 Extension d'un objet

Quand un module peut être étendu, généralement ses types et ses interfaces peuvent être étendus via :

- **Une table d'extension.**
Pour le modèle de données.
- **Une table de méthodes de délégation.**
Pour le modèle de traitements.

Ces deux tables sont systématiquement gérées si le module est considéré comme un type hérité et optionnellement gérées si le module est considéré comme une interface implémentée activée. Ce principe est utilisé par **Up ! 5GL** et **Up ! Compiler**.

Voici les **Application Program Interfaces (API)** pour gérer les extensions :

- **UpsSys.AllouerExtension.**
Alloue l'extension d'un objet en fonction des options activées.
- **UpsSys.LibererExtension.**
Libère l'extension d'un objet en fonction des options activées.

5.2 Gestion de la mémoire

5.2.1 Allocateurs de mémoire

Au lieu d'allouer un à un des objets de taille fixe, il est préférable d'employer un allocateur qui gère les allocations physiques des emplacements mémoire par bloc. Au fur et à mesure que les éléments sont libérés, les blocs le sont de même.

Voici les **Application Program Interfaces (API)** :

- **UpsSys.AjouterAllocateurBloc.**
Crée un nouvel allocateur identifié par un numéro unique.
- **UpsSys.AllouerElementBloc.**
Alloue un nouvel élément à l'allocateur. Ceci est équivalent à un **malloc**.
- **UpsSys.LibererElementBloc.**
Libère un élément précédemment alloué à l'allocateur. Ceci est équivalent à un **free**.
- **UpsSys.LibererAllocateurBloc.**
Détruit un allocateur. Tous les objets encore alloués sont libérés.

5.2.2 Table d'objets

Au lieu de parcourir des listes pour retrouver un objet identifié par un numéro unique, il est préférable d'employer une table qui gère l'accès direct et non séquentiel. La table est étendue au fur et à mesure des besoins.

Eventuellement, les objets de la table sont alloués via un allocateur dédié.

Voici les **Application Program Interfaces (API)** :

- **UpsSys.InitialiserTable.**
Crée une nouvelle table identifiée par un descripteur de type **TypUpsSysTable**.

| | | |
|--|--|---|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

- **UpsSys.AllouerElementTable.**
Alloue un nouvel élément à la table. Ceci est équivalent à un **malloc**.
- **UpsSys.LibererElementTable.**
Libère un élément précédemment alloué à la table. Ceci est équivalent à un **free**.
- **UpsSys.LibererTable.**
Détruit une table. Tous les objets encore alloués sont détruits via le destructeur passé en paramètre.

5.3 Gestion des flux

Cette section décrit comment la gestion des flux est réalisée par **Up ! System**.

Les différentes alternatives possibles sont les suivantes :

- Utiliser les flux du **run-time** du langage **C**.
- Utiliser un mécanisme de flux propriétaire.
La seconde alternative a été retenue pour la raison suivante :
- **Limitation des flux du run-time du langage C.**
Ils s'appliquent uniquement aux fichiers et aux chaînes de caractères. Le mécanisme n'est pas ouvert pour être étendu à d'autres supports tels les socquettes.
- **Sûreté de fonctionnement.**
Ces **Application Program Interfaces (API)** ne comportent pas de mécanisme suffisant visant à contrôler la sûreté de fonctionnement.
- **Portabilité.**
Ces **Application Program Interfaces (API)** ne sont pas complètement portables dans leur sémantique.

Compte-tenu de la première raison, il n'est pas possible de changer de choix étant donné de la montée en puissance d'**Internet** et de la nécessité de l'aisance de programmation pour ces technologies mouvantes.

5.3.1 Principe

Les flux sont des services de haut niveau de lecture et d'écriture d'informations sur un support physique et indépendamment de celui-ci.

Les supports gérés à ce jour sont :

- **Les chaînes de caractères.**
- **Les fichiers.**
- **La communication via Up ! Network.**
Socquettes d'**Up ! Tcp**, tuyau nommé d'**Up ! Named Pipes** ou d'**Up ! Microsoft Network**.

Un flux est modélisé par le type **TypUpsVmFlux**. Les services de haut niveau reposent sur les fonctions de bas niveau suivantes, que le support doit mettre en oeuvre au travers de l'interface **TypUpsVmMethodesFlux** :

- **DeplacerPositionPointeur.**
Déplace la position actuelle de lecture et d'écriture dans le support.
- **Ecrire.**
Ecrit physiquement un paquet d'octets.

| | | |
|--|--|---|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

- **Fermer.**
Ferme le support physique.
- **Lire.**
Lit physiquement un paquet d'octets.
- **LirePositionPointeur.**
Lit la position actuelle de lecture et d'écriture dans le support physique.
- **Ouvrir.**
Ouvre physiquement le support physique.
- **Vider.**
Vide les tampons d'écriture dans le support physique.

5.3.2 Propriétés d'un flux

Les services de haut niveau énuméré prennent en compte les caractéristiques suivantes des flux :

- **Une page de code.**
Pour pouvoir faire les conversions vers ou depuis **Unicode 2.0**.
- **Un mode de lecture et d'écriture.**
Il est soit caractère ou soit binaire.
- **Un mode bloquant ou non.**
Pour les opérations de lecture ou d'écriture.
- **Un mode synchrone ou asynchrone.**
Pour paralléliser les opérations.
- **Une utilisation de tampons.**
Pour servir de cache ou non.

5.3.3 Méthodes d'un flux

Voici les services de haut niveau offert par un flux :

- **FluxAnalyserChaine.**
Lit une chaîne formatée et analyse les champs de la chaîne. La valeur de retour est le nombre de champs décodés.
- **FluxDeplacerPositionPointeur.**
Déplacer la position de lecture et d'écriture courante.
- **FluxEcrireCaracteres.**
Ecrit un paquet de caractères.
- **FluxEcrireOctets.**
Ecrit un paquet d'octets.
- **FluxEcrireUnCaractere.**
Ecrit un caractère.
- **FluxEcrireUnOctet.**
Ecrit un octet.
- **FluxFermer.**
Ferme le flux.

| | | |
|--|--|-----------------------------------|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

- **FluxFormaterChaine.**
Formate une chaîne composée de champs.
- **FluxLireCaracteres.**
Lit un paquet de caractères.
- **FluxLireOctets.**
Lit un paquet d'octets.
- **FluxLirePositionPointeur.**
Lit la position de lecture et d'écriture courante.
- **FluxLireUneLigne.**
Lit une ligne se terminant par un caractère **saut de ligne** `\n`.
- **FluxLireUnCaractere.**
Lit un caractère.
- **FluxLireUnCaractereText.**
Lit un caractère en mode texte même si le flux a été ouvert en mode binaire.
- **FluxLireUnOctet.**
Lit un octet.
- **FluxRejeterUnCaractere.**
Rejette un caractère dans le flux.
- **FluxVider.**
Vide les tampons d'écriture du flux.

Les prototypes de ces **Application Program Interfaces (API)** sont décrits dans le fichier **upssys.e**.

Concernant les paramètres d'**AnalyserChaine** et de **FormaterChaine**, ils sont toujours passés par adresse. Le paramètre **Filtre** peut comporter les jokers suivants :

| Joker. | Sémantique. | Type de données |
|--------|---------------------------------|------------------------|
| %c. | Un caractère. | TypUpsVmUnicode. |
| %hd. | Un entier court signé. | TypUpsVmShort. |
| %hu. | Un entier court non signé. | TypUpsVmUnsignedShort. |
| %hx. | Un entier court en hexadécimal. | TypUpsVmShort. |
| %ld. | Un entier long signé. | TypUpsVmLong. |
| %lf. | Un réel. | TypUpsVmDouble. |
| %lu. | Un entier long non signé. | TypUpsVmUnsignedLong. |
| %lx. | Un entier long en hexadécimal. | TypUpsVmLong. |
| %s. | Une chaîne de caractères. | TypUpsVmUnicode *. |

Tableau 2 – Filtre des API AnalyserChaine et FormaterChaine

5.3.4 Mode d'ouverture d'un flux

Un mode d'ouverture d'un flux est une chaîne de caractères de type **TypUpsVmUnicode *** respectant la sémantique suivante :

| Masques. | Sémantique. |
|----------|-------------|
|----------|-------------|

| | | |
|---|--|---|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |

Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc

| | |
|-------|---|
| "ab" | Ouverture en mode binaire et en écriture. Si le fichier existe déjà, l'écriture se fait à la suite de son contenu. |
| "at" | Ouverture en mode texte et en écriture. Si le fichier existe déjà, l'écriture se fait à la suite de son contenu. |
| "a+b" | Ouverture en mode binaire et en lecture / écriture. Si le fichier existe déjà, l'écriture se fait à la suite de son contenu. |
| "a+t" | Ouverture en mode texte et en écriture. Si le fichier existe déjà, l'écriture se fait à la suite de son contenu. |
| "rb" | Ouverture en mode binaire et en lecture. Le fichier doit exister. |
| "rt" | Ouverture en mode texte et en lecture. Le fichier doit exister. |
| "r+b" | Ouverture en mode binaire et en lecture / écriture. Le fichier doit exister. Le pointeur du fichier est initialisé au début de celui-ci. |
| "r+t" | Ouverture en mode texte et en lecture / écriture. Le fichier doit exister. Le pointeur du fichier est initialisé au début de celui-ci. |
| "wb" | Ouverture en mode binaire et en écriture. Si le fichier existe déjà, son contenu est effacé. |
| "wt" | Ouverture en mode texte et en écriture. Si le fichier existe déjà, son contenu est effacé. |
| "w+b" | Ouverture en mode binaire et en lecture / écriture. Si le fichier existe déjà, son contenu est effacé. |
| "w+t" | Ouverture en mode texte et en lecture / écriture. Si le fichier existe déjà, son contenu est effacé. |

Tableau 3 – Codification des masques d'ouverture de flux

Le mode d'ouverture est employé dans l'**Application Program Interface (API) Ouvrir** déclarée par exemple dans l'interface relative au système d'exploitation.

5.3.5 Exemple d'emploi des flux

Voici un exemple d'emploi des flux :

```

TypUpsVmFlux Descripteur;

(*MonModMachineInterfaceUpsSys->SystemeDeFichiersInterface->Ouvrir)
(Session, &Descripteur, "/tmp/essai.txt", "wt", -1);
(*MonModMachineInterfaceUpsSys->FluxEcrireCaracteres)
(Session, &Descripteur, "Bonjour le monde\n");
(*MonModMachineInterfaceUpsSys->FluxFermer)(Session, &Descripteur);

```

Texte 4 – Exemple d'emploi des flux

5.4 Gestion des systèmes de fichiers

Cette section décrit comment la gestion des systèmes de fichiers est réalisée par **Up! System**.

Les différentes alternatives possibles sont les suivantes :

| | | |
|--|--|---|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

- **Gérer directement le système de fichiers au travers de l'interface avec la machine.**
Intégrer directement les **Application Program Interface (API)** concernant le système de fichiers dans l'interface avec la machine.
 - **Ne pas gérer directement le système de fichiers au travers de l'interface avec la machine.**
Désolidariser les **Application Program Interface (API)** concernant le système de fichiers de celles concernant l'exploitation de la machine.
La seconde alternative a été retenue pour la raison suivante :
 - **Usage d'un système de fichiers transportable.**
Cela permet de créer un système de fichiers transportable parce que le support est un unique fichier physique. L'intérêt est multiple :
 - **Archivage.**
Pour archiver les données des objets métiers gérés par **Up ! Application System** indépendamment de la plate-forme, pour les courriers électroniques, etc.
 - **Lots d'installation.**
Pour constituer les modules d'**Up ! Application System** indépendamment de la plate-forme.
 - **Ressources.**
Pour constituer des ressources graphiques à partir de fichiers élémentaires exploitables par **Up ! Windows.**
 - **Indépendance par rapport au système de fichiers.**
Celui peut être :
 - **Base de données.**
Il s'agit d'un système de fichiers conservés dans une base de données.
 - **Système de fichiers réel.**
Il s'agit du système de fichiers du système d'exploitation.
 - **Système de fichiers virtuel.**
Il s'agit d'un unique fichier du système de fichiers du système d'exploitation.
- Compte-tenu de la nécessité pour les lots d'installation et les ressources graphiques, il n'est pas possible de changer de choix.

5.4.1 Organisation d'un système de fichiers

Un système de fichiers est organisé en répertoires arborescents contenant des fichiers. Le séparateur est le caractère **division /** quelle que soit la plate-forme.

5.4.2 Méthodes d'un système de fichiers

Voici les services de haut niveau offert par un système de fichiers :

- **Pour le système de fichiers stricto sensu.**
 - **ChangerDroitsDAcces.**
Change les droits d'accès à un répertoire ou à un fichier.
 - **CreerUnRaccourci.**
Crée un raccourci pour un répertoire ou pour un fichier.
 - **DémonterUnVolumeAmovible.**
Démonte un volume amovible tel un **CD-Rom**. Il est éjecté si cela est possible.
 - **LireDroitsDAcces.**
Lit les droits d'accès à un répertoire ou à un fichier.

| | | |
|--|--|-----------------------------------|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

- **ListerUnites.**
Liste les unités de stockage disponibles.
- **LireInformations.**
Lit les informations caractérisant un fichier ou un répertoire.
- **MonterUnVolumeAmovible.**
Monte un volume amovible tel un **CD-Rom**.
- **Synchroniser.**
Vide les tampons d'écriture des fichiers vers le support physique.
- Pour les répertoires stricto sensu.
 - **ChangerRepertoire.**
Change de répertoire courant.
 - **CreerRepertoire.**
Crée un nouveau répertoire.
 - **EffacerRepertoire.**
Efface un répertoire déjà existant.
 - **IIExisteRepertoire.**
Retourne **Vrai** s'il existe un répertoire.
 - **LireRepertoireCourant.**
Lit le répertoire courant.
 - **ListerFichiers.**
Liste les fichiers d'un répertoire.
 - **RenommerRepertoire.**
Renomme un répertoire en un autre, en le déplaçant au besoin.
- Pour les fichiers stricto sensu.
 - **EffacerFichier.**
Efface un fichier existant.
 - **IIExisteFichier.**
Retourne **Vrai** s'il existe un fichier.
 - **Ouvrir.**
Ouvre un fichier en lecture et / ou en écriture pour obtenir un flux.
 - **RenommerFichier.**
Renomme un fichier existant, en le déplaçant au besoin.
 - **TraduireFichier.**
Traduit les chemins d'accès au fichier pour tenir compte de la représentation physique du support physique.

Les prototypes de ces **Application Program Interfaces (API)** sont décrits dans le fichier **upssys.e**.

5.5 Gestion de la console

Cette section décrit comment la gestion la console est réalisée par **Up! System**.

Les différentes alternatives possibles sont les suivantes :

| | | |
|--|--|---|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

- **Gérer directement la console au travers de l'interface avec la machine.**
Appeler directement les **Application Program Interface (API)** concernant la console dans l'interface avec la machine.
 - **Ne pas gérer directement la console au travers de l'interface avec la machine.**
Désolidariser l'usage de la console de son implémentation réelle.
La seconde alternative a été retenue pour la raison suivante :
 - **Indépendance par rapport à la console.**
Cela permet d'avoir une souplesse en terme d'architecture en fonction, d'une part, du potentiel de la plate-forme, et, d'autre part, de la politique de déploiement :
 - **Console en mode caractère.**
Il s'agit de la console correspondant au **shell** du système d'exploitation.
 - **Console graphique.**
Il s'agit d'un simple frontal comme celui actuellement proposé ou d'une console embarquée dans une application graphique.
 - **Console avec protocole.**
Il s'agit d'interagir à distance avec le programme via un protocole tel **Telnet** ou **Http**.
- Compte-tenu de la nécessité d'architecture, il n'est pas possible de changer de choix.

5.5.1 Interface de la console

L'interface avec la console correspond au type **GestionConsole** regroupant les méthodes suivantes :

- **AttenteActive.**
Boucle d'attente permettant à la console d'être autonome dans son fonctionnement, notamment pour les rafraîchissements d'écran graphique.
- **ClavierLire.**
Lecture d'une chaîne de caractères **Unicode** depuis la console.
- **EcranEcrire.**
Ecriture d'une chaîne de caractères **Unicode** sur la console.
- **EffacerEcran.**
Effacement de l'écran de la console.
- **Imprimer.**
Impression d'un fichier spoulé sur l'imprimante relié à la console.

5.5.2 Gestionnaire de la console

Le gestionnaire de la console est hébergé dans un module spécialisé, par exemple **Up ! Windows**. Ce dernier est déclaré au moyen de l'**Application Program Interface (API)** **UpsSys.ReferencerConsole** dès qu'elle est disponible. Sinon la console par défaut en mode caractère est utilisée.

Etant donné que cette console doit être disponible le plus tôt possible, il est envisageable que le module soit chargé avant que les services d'**Up ! Virtual Technical Machine** soient tous disponibles. Aussi, les deux méthodes suivantes permettent de parfaire l'initialisation et la terminaison du module de la console :

- **InitialiserEncapsulationKernel.**
Déclaration des objets une fois que **Up ! Kernel** soit disponible.

| | | |
|--|--|-----------------------------------|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

- **TerminerEncapsulationKernel.**
Destruction des objets avant que *Up ! Kernel* ne soit indisponible.

5.6 Interface avec la machine

Cette section décrit comment l'interface avec la machine physique est réalisée pour *Up ! System*.

- Pour les fonctions équivalentes à *Posix*.
 - **DateEnSecondes.**
Retourne la date exprimée en secondes écoulées depuis le 1^{er} janvier 1970.
 - **DateEtHeure.**
Retourne la date et l'heure courante.
 - **DecoderLong.**
Permute les octets d'un *TypUpsVmLong*.
 - **DecoderShort.**
Permute les octets d'un *TypUpsVmShort*.
 - **DecoderUnsignedLong.**
Permute les octets d'un *TypUpsVmUnsignedLong*.
 - **DecoderUnsignedShort.**
Permute les octets d'un *TypUpsVmUnsignedShort*.
 - **LibererMemoire.**
Libère une zone de mémoire privée au processus allouée dynamiquement.
 - **RetourInterAppel.**
Effectue un débranchement inter appel à la manière de `longjmp`.
- Pour les fonctions de gestion du contexte d'exécution.
 - **ArreterProgramme.**
Arrêt le programme courant.
 - **ArreterSysteme.**
Arrête la machine.
 - **ChangerEnvironnement.**
Ecrit la valeur d'une variable environnement ou une clé de registre. Si la valeur est nulle, elle est supprimée.
 - **FermerSession.**
Ferme la session de l'utilisateur.
 - **LireCodeErreur.**
Lit le code erreur et le message associé du dernier appel système exécuté.
 - **LireEnvironnement.**
Lit la valeur d'une variable environnement ou une clé de registre.
 - **LireInformationsSystemeDExploitation.**
Lit les informations caractérisant le système d'exploitation.
 - **LireNomGroupe.**
Lit le nom du groupe de travail de l'utilisateur courant.

| | | |
|--|--|---|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

- ***LireNomUtilisateur.***
Lit le nom de l'utilisateur courant.
- ***LireNumeroUtilisateur.***
Lit le numéro d'identification de l'utilisateur courant.
- ***LirePageDeCodeSysteme.***
Lit la dénomination littérale de la page de codes système d'exploitation.
- ***RelancerProgramme.***
Arrête le programme courant et redémarre-le.
- ***RelancerSysteme.***
Arrête le système et redémarre-le.
- ***RepertoireInstallationSysteme.***
Lit le répertoire d'installation du système.
- Pour les fonctions de gestion des tâches.
 - ***ArreterProcessus.***
Arrête un processus en cours d'exécution sur la machine courante.
 - ***ArreterService.***
Arrête un service sur la machine courante.
 - ***ArreterThread.***
Arrête un thread du processus courant.
 - ***Attendre.***
Suspend l'exécution du processus courant le temps d'un nombre déterminé de secondes.
 - ***ChangerPrioriteProcessus.***
Change la priorité du processus courant.
 - ***ChercherService.***
Cherche un service sur la machine courante.
 - ***CommuterThreads.***
Bascule l'exécution d'un thread à l'autre.
 - ***CreerThread.***
Crée un nouveau thread dans le processus courant.
 - ***DemarrerService.***
Démarré un service sur la machine courante.
 - ***Executer.***
Exécute un programme, ce qui créera un nouveau processus.
 - ***GererSignaux.***
Gère les signaux pour le thread courant.
 - ***LireCodeRetourProcessus.***
Lire le code retour d'un processus terminé sur la machine courante.
 - ***LireEtatSuspensionThread.***
Lire l'état de suspension d'un thread du processus courant.
 - ***LireNumeroProcessus.***
Lire le numéro d'identification du processus courant.

| | | |
|--|--|---|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

- ***LirePrioriteProcessus.***
Lire la priorité du processus courant.
- ***ListerProcessus.***
Liste les processus en cours d'exécution sur la machine courante.
- ***LireTempsProcessus.***
Lire les temps d'exécution d'un processus.
- ***ReprendreExecutionThread.***
Reprend l'exécution d'un thread du processus courant.
- ***SuspendreExecutionThread.***
Suspend l'exécution du thread courant.
- ***ThreadEstEnVie.***
Retourne **Vrai** si un thread particulier est en vie dans le processus courant.
- Pour les fonctions de gestion des verrous.
 - ***DebuterSectionCritique.***
Entre dans une section critique bien déterminée.
 - ***TerminerSectionCritique.***
Sort d'un section critique bien déterminée.
- Pour les flux standard.
 - ***LireEntreeStandard.***
Flux correspondant à l'entrée standard au clavier.
 - ***LireErreurStandard.***
Flux correspondant à la sortie standard en erreur.
 - ***LireSortieStandard.***
Flux correspondant à la sortie standard à l'écran.
- Pour les fonctions spéciales.
 - ***Invoquer.***
Appel d'un traitements avec un nombre variables de paramètres de type générique **UpsSysValeurC** retournant éventuellement un résultat de type générique **UpsSysValeurC**. La convention d'appel peut être l'une des valeurs de l'énuméré **SorteAppelC** :
 - ***CDecl.***
L'appelant empile les paramètres en commençant par le dernier et en terminant par le premier. L'appelant dépile les paramètres.
 - ***CDeclPascal.***
L'appelant empile les paramètres en commençant par le premier et en terminant par le dernier. L'appelant dépile les paramètres.
 - ***StdCall.***
L'appelant empile les paramètres en commençant par le dernier et en terminant par le premier. L'appelé dépile les paramètres.
 - ***StdCallPascal.***
L'appelant empile les paramètres en commençant par le premier et en terminant par le dernier. L'appelé dépile les paramètres.

Les prototypes de ces **Application Program Interfaces (API)** sont décrits dans le fichier **upssys.e**.

| | | |
|--|--|---|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

5.6.1 Section critique

Une section critique permet d'assurer qu'un seul thread exécutera une portion de code en exclusivité de la sorte à maintenir un ensemble de données cohérentes. Voici les sections critiques d'*Up ! Application System* :

| Section critique. | Module. | Sémantique. |
|---------------------------|--|--|
| SC_Alarme. | <i>Up ! System.</i> | Gestion des alarmes. |
| SC_Contrainte. | <i>Up ! Kernel.</i> | Gestion des contraintes. |
| SC_EnregistrementJournal. | <i>Up ! Object Management System.</i> | Gestion des enregistrements dans le journal. |
| SC_Exception. | <i>Up ! Kernel.</i> | Gestion des exceptions. |
| SC_GroupeDeTravail. | <i>Up ! System.</i> | Gestion des groupes de travail. |
| SC_MemoirePartagee. | <i>Up ! System.</i> | Gestion des descripteurs des mémoires partagées. |
| SC_Mutex. | <i>Up ! System.</i> | Gestion des descripteurs des mutex. |
| SC_Semaphore. | <i>Up ! System.</i> | Gestion des descripteurs des sémaphores. |
| SC_Thread. | <i>Up ! System.</i> | Gestion des descripteurs des threads et des processus. |
| SC_Transaction. | <i>Up ! Object Management System.</i> | Gestion des transactions. |
| SC_Ups5GL. | <i>Up ! Fifth Generation Language – Basic.</i> | Gestion des listes chaînées. |
| SC_UpsAna. | <i>Up ! Analyzer.</i> | Gestion des listes chaînées. |
| SC_UpsLg1. | <i>Up ! Fifth Generation Language – Level 1.</i> | Gestion des listes chaînées. |
| SC_UpsMod. | <i>Up ! Module.</i> | Gestion des listes chaînées. |
| SC_UpsNet. | <i>Up ! Network.</i> | Gestion des listes chaînées. |
| SC_UpsNls. | <i>Up ! Natural Language Support.</i> | Gestion des listes chaînées. |
| SC_UpsOrb. | <i>Up ! Object Request Broker.</i> | Gestion des listes chaînées. |
| SC_UpsWin. | <i>Up ! Windows.</i> | Gestion des listes chaînées. |

Tableau 5 – Section critique d'Up ! Virtual Technical Machine

5.6.2 Services pour Up ! Object Management System

L'interface avec la machine d'*Up ! System* comporte les services particuliers suivants pour *Up ! Object Management System* :

- **AllouerMemoirePartagee.**
Alloue ou retrouve une zone de mémoire partagée entre plusieurs processus.
- **AllouerMutex.**
Alloue un nouveau mutex pour synchroniser les threads du processus courant.

| | | |
|--|--|---|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

- **AllouerSemaphore.**
Alloue un nouveau sémaphore pour synchroniser les threads de plusieurs processus.
 - **LibererMemoirePartagee.**
Libère une zone de mémoire partagée déjà existante.
 - **LibererMutex.**
Libère un mutex déjà existant.
 - **LibererRessourcesCritiques.**
Lors de la fermeture de la session d'une tâche, libère toutes les ressources critiques que détenait encore cette tâche.
 - **LibererSemaphore.**
Libère un sémaphore déjà existant.
 - **PrendreMutex.**
Prends le mutex au détriment des autres threads du processus courant.
 - **PrendreJetonSemaphore.**
Prends le jeton du sémaphore au détriment des autres threads des autres processus.
 - **RemettreMutex.**
Lâche le mutex pour les autres threads du processus courant.
 - **RemettreJetonSemaphore.**
Lâche le jeton du sémaphore pour les autres threads des autres processus.
 - **TransmettreAffiliationSemaphoreEtMutex.**
Lors du passage de la connexion interne à une connexion explicite, transmet l'affiliation des sémaphores et des mutex attachés à la connexion interne à la connexion explicite.
- Les prototypes de ces **Application Program Interfaces (API)** sont décrits dans le fichier **upssys.e**.

| | | |
|--|--|-----------------------------------|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

6 Modèle de données

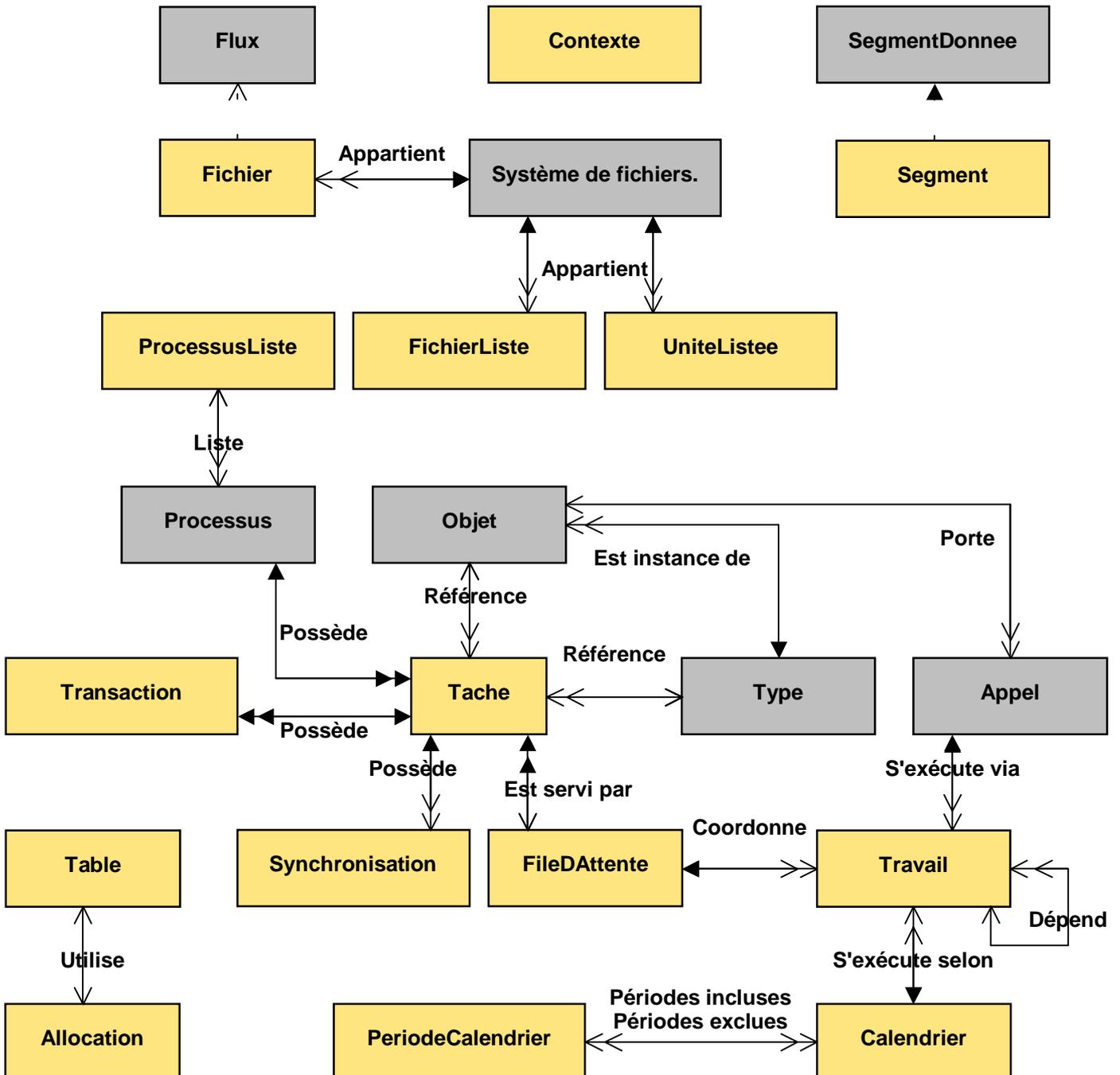


Diagramme 6 – Modèle physique des données publiques du module *Up ! System*

| | | |
|--|--|--|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

Toutes les entités suivantes sont décrites dans le fichier **upssys.e** :

| Entité. | Description. |
|-----------------------------|--|
| Allocateur. | Allocateur d'objets en bloc. |
| Appel. | Modélisation d'un objet appel d'une procédure, d'une fonction ou d'une méthode. |
| Calendrier. | Calendrier d'exécution d'un travail. |
| Contexte. | Description d'un ensemble de paramètres d'un module du programme. |
| Fichier. | Modélisation d'un fichier. |
| FichierListe. | Résultat du listage du contenu d'un répertoire. |
| FileDAttente. | File d'attente de travaux à exécuter. |
| Flux. | Flux de lecture ou d'écriture d'informations. Cette définition est dans upsvm.e |
| Objet. | Modélisation d'un objet générique. |
| PeriodeCalendrier. | Période de disponibilité ou d'indisponibilité d'un calendrier. |
| Processus. | Processus s'exécutant sur la machine. |
| ProcessusListe. | Résultat du listage des processus s'exécutant sur la machine. |
| Segment. | Modélisation d'un segment. |
| Synchronisation. | Synchronisation posée par une tâche. |
| Système de fichiers. | Système de fichiers mettant en oeuvre l'interface TypUpsSysSystemeDeFichiersInterface . |
| Table. | Table d'accès direct à un ensemble d'objets. |
| Tache. | Tâche exécutée par un thread qui est identifiée par une session. |
| Transaction. | Modélisation d'un objet transaction. |
| Travail. | Travail à exécuter. |
| Type. | Modélisation d'un objet type. |
| UniteListee. | Résultat du listage des unités de stockage du système d'exploitation. |

Tableau 7 – Glossaire du modèle physique des données publiques du module **Up ! System**

| | | |
|--|--|-----------------------------------|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

7 Composants techniques

Le module **Up ! System** pour le projet **Up ! Application System** est constitué des composants suivants :

| | |
|--|-----------------|
| Fichiers du module. | |
| <ul style="list-style-type: none"> Fichier upssys.e – Définition des interfaces des bibliothèques gérées par Up ! System. Fichier upssys.h – En-tête privé de Up ! System. Fichier upssys.def – Exportation des symboles pour Windows. Fichier upssys_nt.e – Définition de l'adaptateur des APIs natives utilisées par Up ! System. Fichier upssys_nt.h – En-tête privé de l'adaptateur des APIs natives utilisées par Up ! System. Fichier upssys_nt.def – Exportation des symboles pour Windows. | |
| Composants. | upssys0. |
| Description. | |
| Interface entre Up ! System et Up ! Module . | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier upssys0.cpp – Fichier source. Fichier upssys0.h – En-tête privé de upssys0.cpp. Fichier upssys0.e – En-tête protégé de upssys0.cpp. | |
| Composants. | upssys1. |
| Description. | |
| <ul style="list-style-type: none"> API de gestion de la mémoire. API de gestion des conversions. Gestion des flux. Flux sur les chaînes de caractères. Gestion des alarmes. Gestion des extensions des modules. Fonctions partagées par les adaptateurs. | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier upssys1.cpp – Fichier source. Fichier upssys1.h – En-tête privé de upssys1.cpp. Fichier upssys1.e – En-tête protégé de upssys1.cpp. | |
| Composants. | upssys2. |
| Description. | |
| Gestion du contexte d'exécution du programme : | |
| <ul style="list-style-type: none"> Paramètres. Fichier d'initialisation. Variable environnement. | |
| Fichiers. | |

| | | |
|--|--|-----------------------------------|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

- Fichier **upssys2.cpp** – Fichier source.
- Fichier **upssys2.h** – En-tête privé de **upssys2.cpp**.
- Fichier **upssys2.e** – En-tête protégé de **upssys2.cpp**.

| | |
|--------------------|------------------|
| Composants. | upssys10. |
|--------------------|------------------|

Description.

- Méthode du type **FichierListe**.
- Méthode du type **ProcessusListe**.
- Méthode du type **UniteListee**.
- Procédure et fonctions de **upssys.upi**.

Fichiers.

- Fichier **upssys10.cpp** – Fichier source.
- Fichier **upssys10.h** – En-tête privé de **upssys10.cpp**.
- Fichier **upssys10.e** – En-tête protégé de **upssys10.cpp**.

| | |
|--------------------|------------------|
| Composants. | uspsys11. |
|--------------------|------------------|

Description.

Méthodes du type **Tache**.

Fichiers.

- Fichier **upssys11.cpp** – Fichier source.
- Fichier **upssys11.h** – En-tête privé de **upssys11.cpp**.
- Fichier **upssys11.e** – En-tête protégé de **upssys11.cpp**.

| | |
|--------------------|------------------|
| Composants. | upssys12. |
|--------------------|------------------|

Description.

Méthodes du type **Synchronisation**.

Fichiers.

- Fichier **upssys12.cpp** – Fichier source.
- Fichier **upssys12.h** – En-tête privé de **upssys12.cpp**.
- Fichier **upssys12.e** – En-tête protégé de **upssys12.cpp**.

| | |
|--------------------|------------------|
| Composants. | upssys13. |
|--------------------|------------------|

Description.

Méthodes du type **FileDAttente**.

Fichiers.

- Fichier **upssys13.cpp** – Fichier source.
- Fichier **upssys13.h** – En-tête privé de **upssys13.cpp**.
- Fichier **upssys13.e** – En-tête protégé de **upssys13.cpp**.

| | |
|--------------------|------------------|
| Composants. | upssys14. |
|--------------------|------------------|

Description.

Méthodes du type **Travail**.

Fichiers.

| | | |
|---|--|-----------------------------------|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |

Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc

| | |
|---|------------------|
| <ul style="list-style-type: none"> Fichier upssys14.cpp – Fichier source. Fichier upssys14.h – En-tête privé de upssys14.cpp. Fichier upssys14.e – En-tête protégé de upssys14.cpp. | |
| Composants. | upssys15. |
| Description. | |
| <ul style="list-style-type: none"> Méthode de l'interface Flux. Méthodes du type Fichier. | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier upssys15.cpp – Fichier source. Fichier upssys15.h – En-tête privé de upssys15.cpp. Fichier upssys15.e – En-tête protégé de upssys15.cpp. | |
| Composants. | upssys16. |
| Description. | |
| Méthodes du type Segment . | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier upssys16.cpp – Fichier source. Fichier upssys16.h – En-tête privé de upssys16.cpp. Fichier upssys16.e – En-tête protégé de upssys16.cpp. | |
| Composants. | upssys17. |
| Description. | |
| Méthodes du type PeriodeCalendrier . | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier upssys17.cpp – Fichier source. Fichier upssys17.h – En-tête privé de upssys17.cpp. Fichier upssys17.e – En-tête protégé de upssys17.cpp. | |
| Composants. | upssys18. |
| Description. | |
| Méthodes du type Calendrier . | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier upssys18.cpp – Fichier source. Fichier upssys18.h – En-tête privé de upssys18.cpp. Fichier upssys18.e – En-tête protégé de upssys18.cpp. | |
| Composants. | upssys99. |
| Description. | |
| Interface entre Up ! System et Up ! Kernel . | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier upssys99.cpp – Fichier source. Fichier upssys99.h – En-tête privé de upssys99.cpp. Fichier upssys99.e – En-tête protégé de upssys99.cpp. | |

| | | |
|--|--|-----------------------------------|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

| | |
|--|--------------------|
| Composants. | sys390. |
| Description. | |
| Adaptateur pour Os 390. | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier sys390.cpp – Adaptateur pour Os 390. Fichier sys390.h – En-tête privé de sys390.cpp. Fichier sys390.e – En-tête protégé de sys390.cpp. | |
| Composants. | sys400. |
| Description. | |
| Adaptateur pour Os 400. | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier sys400.cpp – Adaptateur pour Os 400. Fichier sys400.h – En-tête privé de sys400.cpp. Fichier sys400.e – En-tête protégé de sys400.cpp. | |
| Composants. | sysmac. |
| Description. | |
| Adaptateur pour Macintosh. | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier sysmac.cpp – Fichier source. Fichier sysmac.h – En-tête privé de sysmac.cpp. Fichier sysmac.e – En-tête protégé de sysmac.cpp. | |
| Composants. | sysunx. |
| Description. | |
| Adaptateur pour Unix. | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier sysunx.cpp – Fichier source. Fichier sysunx.h – En-tête privé de sysunx.cpp. Fichier sysunx.e – En-tête protégé de sysunx.cpp. | |
| Composants. | sysw32. |
| Description. | |
| Adaptateur pour Windows. | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier sysw32.cpp – Fichier source. Fichier sysw32.h – En-tête privé de sysw32.cpp. Fichier sysw32.e – En-tête protégé de sysw32.cpp. | |
| Composants. | upssys0_nt. |
| Description. | |
| Interface entre l'adaptateur des APIs natives utilisées par Up ! System et Up ! Module. | |

| | | |
|--|--|-----------------------------------|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

| | |
|---|-------------------|
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier upssys0_nt.cpp – Fichier source. Fichier upssys0_nt.h – En-tête privé de upssys0_nt.cpp. Fichier upssys0_nt.e – En-tête protégé de upssys0_nt.cpp. | |
| Composants. | sys390_nt. |
| Description. | |
| Adaptateur des APIs natives pour Os 390 utilisées par Up ! System . | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier sys390_nt.cpp – Adaptateur pour Os 390. Fichier sys390_nt.h – En-tête privé de sys390.cpp. Fichier sys390_nt.e – En-tête protégé de sys390.cpp. | |
| Composants. | sys400_nt. |
| Description. | |
| Adaptateur des APIs natives pour Os 400 utilisées par Up ! System . | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier sys400_nt.cpp – Adaptateur pour Os 400. Fichier sys400_nt.h – En-tête privé de sys400_nt.cpp. Fichier sys400_nt.e – En-tête protégé de sys400_nt.cpp. | |
| Composants. | sysi86_nt. |
| Description. | |
| Adaptateur des APIs natives pour Intel x86 utilisées par Up ! System écrites en assembleur | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier sysi86_nt.cpp – Fichier source. Fichier sysi86_nt.h – En-tête privé de sysi86_nt.cpp. Fichier sysi86_nt.e – En-tête protégé de sysi86_nt.cpp. | |
| Composants. | sysmac_nt. |
| Description. | |
| Adaptateur des APIs natives pour Macintosh utilisées par Up ! System . | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier sysmac_nt.cpp – Fichier source. Fichier sysmac_nt.h – En-tête privé de sysmac_nt.cpp. Fichier sysmac_nt.e – En-tête protégé de sysmac_nt.cpp. | |
| Composants. | sysunx_nt. |
| Description. | |
| Adaptateur des APIs natives pour Unix utilisées par Up ! System . | |
| Fichiers. | |

| | | |
|--|--|--|
|  | Spécification technique du module | Date rédaction : 10 juin 2004. |
| | Diffusion restreinte | Date validation : |
| Référence : UpComp-UpsSys-000003-A Spécification technique du module UpsSys.doc | | |

| | |
|--|-------------------|
| <ul style="list-style-type: none"> Fichier sysunx_nt.cpp – Fichier source. Fichier sysunx_nt.h – En-tête privé de sysunx_nt.cpp. Fichier sysunx_nt.e – En-tête protégé de sysunx_nt.cpp. | |
| Composants. | sysw32_nt. |
| Description. | |
| Adaptateur des APIs natives pour Windows utilisées par Up ! System . | |
| Fichiers. | |
| <ul style="list-style-type: none"> Fichier sysw32_nt.cpp – Fichier source. Fichier sysw32_nt.h – En-tête privé de sysw32_nt.cpp. Fichier sysw32_nt.e – En-tête protégé de sysw32_nt.cpp. | |

Tableau 8 – Composants techniques du module

Fin de document