

**IASI**

**THALES INFORMATION SYSTEMS**

**IA-DD-2100-9564-THA**

Edition : 02 Date : 02/09/2002

Révision : 01 Date : 09/02/2004

MT : X Code diffusion : E

Réf. : -

**DOSSIER DE DEFINITION  
DOSSIER DE DEFINITION DES SERVICES COMMUNS DU  
LOGICIEL DE L'OPS IASI**

<b>Rédigé par :</b> CABANE Philippe	THALES IS	le :	
<b>Validé par :</b> AYER Patrick BRANET Pascal	THALES IS THALES IS	le :	
<b>Pour application :</b> MORENO Richard	DTS/MID/VM/TD	le :	

C N E S	IASI	IA-DD-2100-9564-THA	Edit. : 02	Date : 02/09/2002
			Rév. : 01	Date : 09/02/2004
Référence : -		Page : i.2		

## BORDEREAU D'INDEXATION

CONFIDENTIALITE : NC		MOTS CLES : Conception,Détaillée, Services Communs, CMN, IASI, algorithmes, Traitement d'images			
TITRE DU DOCUMENT : Dossier de Définition Dossier de Définition des Services Communs du Logiciel de l'OPS IASI					
AUTEUR(S) : CABANE Philippe THALES IS					
RESUME : Ce document constitue le dossier de définition du composant des Services Communs (CMN) du logiciel OPS IASI Level 1. Il présente la conception détaillée des modules qui le composent.					
DOCUMENTS RATTACHES : Ce document vit seul.				LOCALISATION :	
VOLUME : 1	NBRE TOTAL DE PAGES : 146146 DONT PAGES LIMINAIRES : 0 NBRE DE PAGES SUPPL. : 0		DOCUMENT COMPOSITE : N		LANGUE : FR
GESTION DE CONF. : F			RESP. GEST. CONF. : GOMEZ MH		
CAUSE D'EVOLUTION : action 124:suppression de la phrase réutilisation à 30% dans le paragraphe 7 Strategie de tests unitaires					
CONTRAT : 01/8937					
SYSTEME HOTE : Microsoft Word 9.0, J:\PROJETS\IASI\Modèles CNES\CnesIndustriel97.dot v2.0.4					

## DIFFUSION INTERNE

Nom	Sigle	BPi	Observations
BLUMSTEIN Denis	DSO/OT/SE/IA	2504	
CHALON Gilles	DSO/OT/SE/IA	2504	
PONCE Ghislaine	DSO/OT/SE/IA	2504	
SEGALEN Barbara	DSO/SG/CS	2504	
DUPLAA Michel	DTS/MID/VM/D	1502	
MARQUIER Henry	DTS/MID/VM/TD	1502	
MORENO Richard	DTS/MID/VM/TD	1502	
GOMEZ Marie-Hélène	DTS/MID/VM/MG	1502	
BAILLY Isabelle	DSO/OT/QTIS/VP	811	
RAYSSIGUIER Michel	DTS/OT/QTIS/VP	811	
MATHIEU Nathalie	DTS/AQ/QIS/SC	1415	
RICHARD Pascal	DEE/IR/ISM/IS	1311	

## DIFFUSION EXTERNE

Nom	Sigle	Observations
AYER Patrick	THALES IS	
BOBIN Serge	THALES IS	
BRANET Pascal	THALES IS	
PASCAL Jean-Luc	THALES IS	
CABANE Philippe	THALES IS	

## MODIFICATION

Ed.	Rév.	Date	Référence, Auteur(s), Causes d'évolution
02	01	09/02/2004	- MASSART Benjamin THALES IS action 124:suppression de la phrase réutilisation à 30% dans le paragraphe 7 Strategie de tests unitaires
02	00	02/09/2002	- CABANE Philippe THALES IS Mise au format GDOC et Rédaction de la CD des Services Communs
01	00	02/09/2002	- CABANE Philippe THALES IS Création du document

## SOMMAIRE

<b>GLOSSAIRE ET LISTE DES PARAMETRES AC &amp; AD .....</b>	<b>9</b>
<b>1. GENERALITES .....</b>	<b>10</b>
1.1. DOCUMENTS DE REFERENCE ET APPLICABLES .....	10
<b>2. INTRODUCTION .....</b>	<b>11</b>
2.1. OBJECTIF .....	11
2.2. DOMAINE D'APPLICATION .....	11
<b>3. PRESENTATION.....</b>	<b>12</b>
3.1. DIAGRAMME DE CONTEXTE.....	12
3.2. RAPPEL DES FONCTIONNALITES .....	13
<b>4. JUSTIFICATIONS TECHNIQUES DE L'ENSEMBLE .....</b>	<b>14</b>
<b>5. SPECIFICATIONS DE L'ENSEMBLE .....</b>	<b>15</b>
<b>6. CONCEPTION DE L'ENSEMBLE.....</b>	<b>16</b>
<b>7. STRATEGIE DE TESTS UNITAIRES .....</b>	<b>17</b>
<b>8. COMPOSITION DE L'ENSEMBLE .....</b>	<b>19</b>
<b>9. DESCRIPTION DES UNITES DE REALISATION .....</b>	<b>22</b>
9.1. CLASSE CFG__CONFIGURATION .....	22
9.1.1. Spécifications .....	22
9.1.2. Conception Détaillée .....	22
9.2. CLASSE CMN__CHAINE .....	25
9.2.1. Spécifications .....	25
9.2.2. Conception Détaillée .....	26
9.3. CLASSE CMN__CMD .....	32
9.3.1. Spécifications .....	32
9.3.2. Conception Détaillée .....	33
9.4. CLASSE CMN__CMDTM.....	33
9.4.1. Spécifications .....	33
9.4.2. Conception Détaillée .....	34
9.5. CLASSE CMN__CONFIG .....	34
9.5.1. Spécifications .....	34
9.5.2. Conception Détaillée .....	35
9.6. CLASSE CMN__CONFIGURATION .....	39
9.6.1. Spécifications .....	39

9.6.2. Conception Détaillée .....	40
9.7. CLASSE CMN__DATE .....	58
9.7.1. Spécifications .....	58
9.7.2. Conception Détaillée .....	60
9.8. CLASSE CMN__FICHER .....	74
9.8.1. Spécifications .....	74
9.8.2. Conception Détaillée .....	75
9.9. CLASSE CMN__PGESERVICES .....	79
9.9.1. Spécifications .....	79
9.9.2. Conception Détaillée .....	80
9.10. CLASSE CMN__PGESERVICESAPI .....	82
9.10.1. Spécifications .....	82
9.10.2. Conception Détaillée .....	83
9.11. CLASSE CMN__PGESERVICESSTUB .....	86
9.11.1. Spécifications .....	86
9.11.2. Conception Détaillée .....	86
9.12. CLASSE CMN__SECURITE .....	89
9.12.1. Spécifications .....	89
9.12.2. Conception Détaillée .....	89
9.13. CLASSE CMN__STAGETM .....	91
9.13.1. Spécifications .....	91
9.13.2. Conception Détaillée .....	91
9.14. CLASSE CMN__TRACE .....	92
9.14.1. Spécifications .....	92
9.14.2. Conception Détaillée .....	93
9.15. CLASSE CMN__VERSION .....	96
9.15.1. Spécifications .....	96
9.15.2. Conception Détaillée .....	96
9.16. CLASSE ERR__EXCEPTION .....	97
9.16.1. Spécifications .....	97
9.16.2. Conception Détaillée .....	97
9.17. CLASSE JDB__CALLBACK .....	97
9.17.1. Spécifications .....	97
9.17.2. Conception Détaillée .....	98
9.18. CLASSE JDB__CLIENT .....	99
9.18.1. Spécifications .....	99
9.18.2. Conception Détaillée .....	99
9.19. CLASSE JDB__MESSAGE .....	101
9.19.1. Spécifications .....	101
9.19.2. Conception Détaillée .....	102
9.20. CLASSE MP__ARRETPROCESS .....	104
9.20.1. Spécifications .....	104
9.20.2. Conception Détaillée .....	104
9.21. CLASSE MP__COMMAND .....	106
9.21.1. Spécifications .....	106
9.21.2. Conception Détaillée .....	106

9.22. CLASSE MP__DUMP .....	108
9.22.1. Spécifications .....	108
9.22.2. Conception Détaillée .....	108
9.23. CLASSE MP__HKTSTATUS .....	109
9.23.1. Spécifications .....	109
9.23.2. Conception Détaillée .....	110
9.24. CLASSE MP__SPACECRAFT .....	112
9.24.1. Spécifications .....	112
9.24.2. Conception Détaillée .....	113
9.25. CLASSE MP__STAGETM .....	114
9.25.1. Spécifications .....	114
9.25.2. Conception Détaillée .....	115
9.26. CLASSE MP__UNPROCESS .....	116
9.26.1. Spécifications .....	116
9.26.2. Conception Détaillée .....	116
9.27. CLASSE SD__INIT .....	117
9.27.1. Spécifications .....	117
9.27.2. Conception Détaillée .....	118
9.28. CLASSE TES__EVENEMENT .....	119
9.28.1. Spécifications .....	119
9.28.2. Conception Détaillée .....	120
9.29. CLASSE CMN_DAT__EVENEMENT .....	121
9.29.1. Spécifications .....	121
9.29.2. Conception Détaillée .....	122
9.30. CLASSE CMN_DAT__LISTEOBJETS .....	123
9.30.1. Spécifications .....	123
9.30.2. Conception Détaillée .....	124
9.31. CLASSE CMN_DAT__MODELEACTION .....	127
9.31.1. Spécifications .....	127
9.31.2. Conception Détaillée .....	127
9.32. CLASSE CMN_DAT__MODELETACHE .....	128
9.32.1. Spécifications .....	128
9.32.2. Conception Détaillée .....	129
9.33. CLASSE CMN_DAT__MODELETRAITEMENT .....	130
9.33.1. Spécifications .....	130
9.33.2. Conception Détaillée .....	131
9.34. CLASSE CMN_DAT__OBJET .....	132
9.34.1. Spécifications .....	132
9.34.2. Conception Détaillée .....	133
9.35. CLASSE CMN_DAT__WORKORDER .....	136
9.35.1. Spécifications .....	136
9.35.2. Conception Détaillée .....	136
9.36. CLASSE CMN_OBS__OBSERVATEUR .....	138
9.36.1. Spécifications .....	138
9.36.2. Conception Détaillée .....	138
9.37. CLASSE CMN_OBS__SUJET .....	139

9.37.1. Spécifications .....	139
9.37.2. Conception Détaillée .....	139
9.38. CLASSE CMN_PRS_DUMPFORMATTARGET .....	140
9.38.1. Spécifications .....	140
9.38.2. Conception Détaillée .....	141
9.39. CLASSE CMN_PRS_ERRORREPORTER .....	141
9.39.1. Spécifications .....	141
9.39.2. Conception Détaillée .....	142
9.40. CLASSE CMN_PRS_PARSER .....	143
9.40.1. Spécifications .....	143
9.40.2. Conception Détaillée .....	143



C N E S	IASI	IA-DD-2100-9564-THA	Edit. : 02	Date : 02/09/2002
			Rév. : 01	Date : 09/02/2004
Référence : -			Page : 9	

## GLOSSAIRE ET LISTE DES PARAMETRES AC & AD

API	Applicative Program Interface
AVHRR	Advanced Very High Resolution Radiometer : radiomètre avancé à très haute résolution (visible et infrarouge) sur les satellites polaires
CGS	Core Ground Segment : segment-sol développé par ALCATEL sous contrat d'EUMETSAT, et dans lequel l'OPS ira s'insérer
Dump	Un dump est une ensemble de données correspondant à un intervalle entre 2 téléchargements. En moyenne il s'agit d'une orbite. Les produits METOP sont structurés autour de cette notion de dump : un produit=un dump
EPS	EUMETSAT Polar System : Système 'Polaire d'EUMETSAT
IASI	Infrared Atmospheric Sounding Interferometer : interferomètre de sondage atmosphérique dans l'infrarouge. L'objet de ce document est de spécifier le logiciel opérationnel du traitement sol des données IASI.
ISRFEM	Instrument Spectral Response Function Estimation Model
MCS	Monitoring and Control Segment
MLA	MCS Local Agent
OPS	Logiciel Opérationnel (Operational Software) : correspond au IASI level 1 PPS dans les glossaires d'EUMETSAT. PPS=Product Processing Software
OS	Operating System
PDS	Payload Data Segment : Centre de Mission ENVISAT
PGF	Product Generation Facility
PPF	Product Processing Facility
SIF	Simulateurs d'Interfaces du F-PAC ENVISAT (sous système du CNES)

**Liste des paramètres AC :**

**Liste des paramètres AD :**

## 1.GENERALITES

### 1.1.DOCUMENTS DE REFERENCE ET APPLICABLES

La liste des documents constituant le référentiel du projet OPS-IASI est détaillée dans la « Liste Unique » du Logiciel OPS-IASI [DR100].

## 2.INTRODUCTION

### 2.1.OBJECTIF

Ce document a pour but de présenter l'ensemble Services Communs ou CMN de l'OPS-IASI.

Le document présente d'abord les justifications techniques qui nous ont amenées à identifier ce composant. Les spécifications techniques du composant sont alors rappelées. Le chapitre suivant présente sa décomposition en paquetages et en unités de réalisation (classes).

Chaque unité de réalisation est présentée suivants 2 axes : la stratégie de test unitaire mise en œuvre et la Conception Détaillée de la classe. Cette conception est extraite des sources eux mêmes, ce qui permet d'assurer très facilement la cohérence entre ce document et ce qui est réellement codé. La conception détaillée de chaque module(classe) est subdivisée en 3 parties : spécification et justification technique, liste des procédures (méthodes) du module (classe), conception détaillée des procédures (méthodes).

Dans le cas de classes réutilisées sans adaptation, la conception détaillée n'est pas mise à jour.

Dans le cas de classes réutilisées avec adaptation, seule la conception détaillée des méthodes nouvelles ou adaptées est mise à jour.

### 2.2.DOMAINE D'APPLICATION

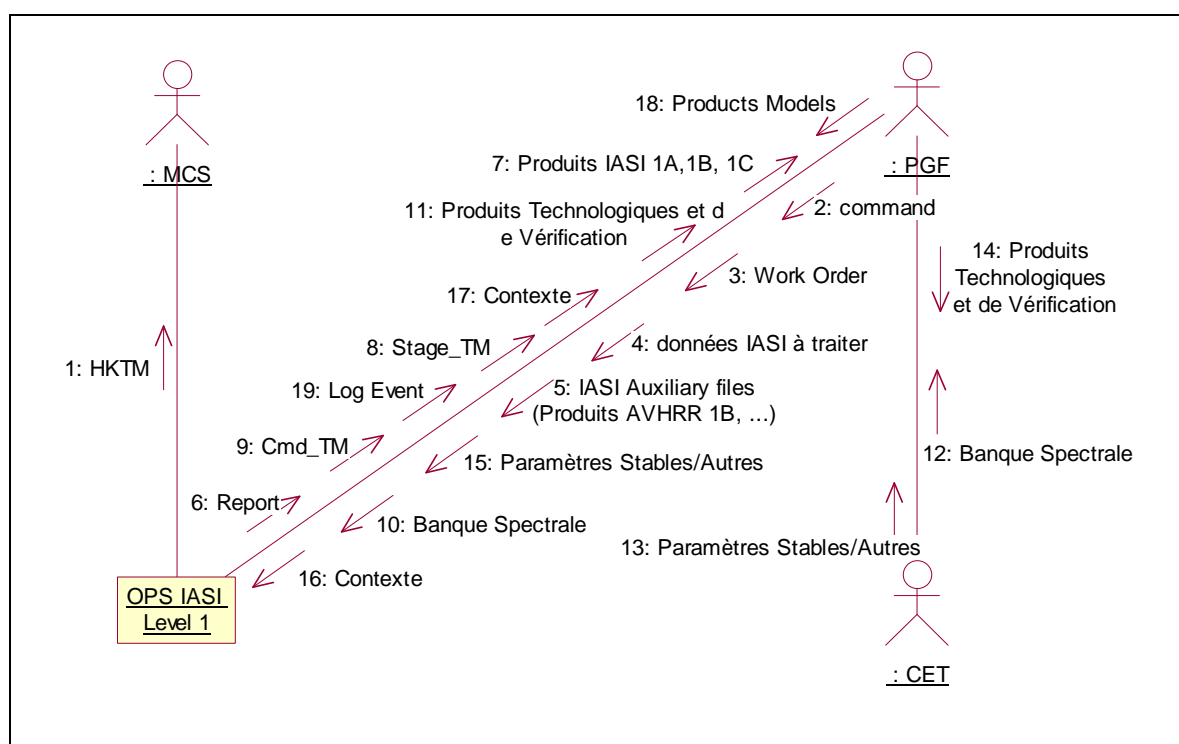
Ce présent document s'applique aux différentes phases de la réalisation du logiciel OPS IASI décrites dans le Plan d'Application [DA102].

Il est livré au CNES en version définitive en fin de Conception Détaillée.

### 3. PRESENTATION

#### 3.1. DIAGRAMME DE CONTEXTE

Le diagramme de contexte présente les interactions de l'OPS avec les systèmes externes à travers les interfaces échangées. Le schéma ci-dessous présente le diagramme de contexte de l'OPS.



**Figure 1 : Diagramme de Contexte de l'OPS-IASI**

L'OPS IASI est interfacé avec 2 autres sous-systèmes du CGS :

- le MCS qui est le système de Monitoring&Control des sous-systèmes du CGS. Le rôle du MCS est de surveiller le fonctionnement et la disponibilité des divers sous-systèmes du CGS. Pour ceci, les sous-systèmes dont l'OPS doivent lui fournir régulièrement ou sur anomalie des informations de suivi : les HK TM statuts (1).
- le PGF qui est le système de Monitoring&Control des chaînes de traitement. Son rôle est de répartir et surveiller les traitements des données des charges utiles EPS en fonction d'un plan de travail sur les calculateurs disponibles. Le PGF a en charge de mettre à disposition l'ensemble des données nécessaires à l'OPS pour effectuer un traitement : work order (3), données à traiter (4), données auxiliaires (Produits AVHRR, ...) (5), Contexte (16), Paramètres Stables/Autres (15), Banque Spectrale (10), Product Models (18) ; puis, de contrôler à travers les commandes

(2) la génération des produits. Le traitement d'une commande par l'OPS donne lieu à l'émission d'un compte rendu : cmde\_TM (9). En fin de traitement, l'OPS génère un rapport (6) et met les données produites à disposition du PGF qui a en charge de les récupérer et de les diffuser : Produits IASI 1A/1B/1C (7), Contexte (17), Produits Technologiques et de Vérification (11). L'OPS prévient le PGF de la disponibilité d'un nouveau produit par l'envoi d'un message Stage\_TM (8). Les événements de l'OPS sont signalés à l'aide de messages Journal de Bord (19).

## 3.2.RAPPEL DES FONCTIONNALITES

L'analyse des exigences de l'OPS met en évidence les principales fonctions suivantes :

- **la fonction de génération des produits IASI L1 :**
  - l'initialisation des données de configuration (Banque Spectrale, ...),
  - l'exécution des algorithmes de la chaîne de traitement,
  - la génération des produits,
- **les services :**
  - la surveillance et le contrôle de l'OPS,
  - la réception et le traitement des work orders,
  - la gestion des messages d'informations,
  - la gestion des anomalies.

## 4.JUSTIFICATIONS TECHNIQUES DE L'ENSEMBLE

Le logiciel OPS a été découpé en 2 ensembles fonctionnels qui permettent de bien isoler l'aspect production des données IASI L1 et algorithmes de l'aspect interfaçage avec les sous-systèmes externes MCS et PGF. Nous avons en plus identifié l'ensemble des Services Communs qui regroupe des bibliothèques soit de services généraux (gestion des répertoires, ...), soit de services utilisés par les 2 ensembles fonctionnels.

Les ensembles de conception détaillée de l'OPS sont les suivants :

- Monitoring & Control : interfaçage avec les sous-systèmes externes,
- Serveur de Données : production des données IASI L1,
- Services Communs.

Ce découpage se retrouve dans l'architecture logicielle du produit :

- Monitoring & Control regroupe les processus MP, JDBS, MSGS, TES, WOM,
- Serveur de Données est implémenté par un unique processus : le SD.

De plus, ce découpage permet de regrouper les traitements algorithmiques au sein d'un unique ensemble : l'ensemble Serveur de Données.

Le tableau suivant réparti les fonctions identifiées au §3.2 dans les ensembles de conception détaillé.

<i>Ensembles</i>	<i>Fonctions</i>
Monitoring & Control	Surveillance et le contrôle de l'OPS Réception et le traitement des work orders Gestion des messages d'informations Gestion des anomalies
Serveur de Données	Initialisation des données de configuration Exécution des algorithmes de la chaîne de traitement Génération des produits

## 5.SPECIFICATIONS DE L'ENSEMBLE

Le tableau suivant répartit les chapitres du Dossier de Spécification [DA104] applicables à chacun des ensembles.

Chapitres des Spécifications Logicielles [DA104]	Applicable à l'Ensemble
3.4.1 : Présentation Générale	Monitoring & Control Serveur de Données Services Communs
3.4.2 : L'OPS IASI	Monitoring & Control Serveur de Données Services Communs
3.4.3 : Contrôler et Surveiller l'OPS-IASI	Monitoring & Control
3.4.4 : Initialiser le Traitement d'un Dump	Serveur de Données
3.4.5 : Générer les Produits IASI Level 1	Serveur de Données
3.4.6 : Pré traitement du Granule	
3.4.6.1 : Traitements des Modes	Serveur de Données Services Communs
3.4.6.2 : Analyse Télémessure et Cas Dégradés	Serveur de Données
3.4.6.3 : Mécanisme de Gestion de l'Overlap	Serveur de Données
3.4.7 : Chaîne Image	Serveur de Données
3.4.8 : Chaîne ISRFEM	Serveur de Données
3.4.9 : Chaîne Produit (spectre)	Serveur de Données
3.4.10 : Chaîne Produit (localisation)	Serveur de Données
3.4.11 : Chaîne Monitoring	Serveur de Données
3.4.12 : Post-traitement du Granule	Serveur de Données
3.4.13 : Monitoring Production	Monitoring & Control
4 : Spécifications Opérationnelles	Monitoring & Control Serveur de Données Services Communs
5.1 : Design et Développement	Monitoring & Control Serveur de Données Services Communs

## 6.CONCEPTION DE L'ENSEMBLE

Le tableau suivant présente les éléments du Dossier de Conception Préliminaire [DA108] desquels dérive les ensembles de conception détaillés.

Chapitres de la Conception Préliminaire [DA108]	Applicable à l'Ensemble
3 : <i>Architecture Logicielle</i>	
3.1 : Le Découpage en Processus	Monitoring & Control Serveur de Données
3.2 : Architecture des Répertoires	Monitoring & Control
3.3 : Les Solutions d'Implémentation	
3.3.1 : La Parallélisation des Traitements	Serveur de Données
3.3.2 : L'Encapsulation	Services Communs Serveur de Données
3.3.3 : Fonctionnement en Standalone	Monitoring & Control
3.3.4 : La Gestion des Traces	Monitoring & Control Serveur de Données
3.4 : La Configuration de l'OPS	
3.4.1 : Présentation	Monitoring & Control Serveur de Données Services Communs
3.4.2 : Fichiers de Paramétrage de la Parallélisation du Traitement	Serveur de Données
4 : <i>Architecture de Classes</i>	
4.1 : L'Architecture Générale	Monitoring & Control Serveur de Données Services Communs
4.2 : Les Process Applicatifs	
4.2.1 : Le Main Process	Monitoring & Control
4.2.2 : Le Work Order Manager	Monitoring & Control
4.2.3 : Le Serveur de Données	Serveur de Données
4.3 : La Couche de Services	
4.3.2 : Le Serveur de Messages (MSGs)	Monitoring & Control
4.3.3 : Le Serveur d'Evénements Timer (TES)	Monitoring & Control
4.3.4 : Le Serveur de Messages JdB (JDBS)	Monitoring & Control
4.3.5 : La Librairie des Services Communs (CMN)	Services Communs



## 7.STRATEGIE DE TESTS UNITAIRES

Plusieurs stratégies, conformes au Plan d'Application [DA012], ont été définies pour tester unitairement les modules/classes du logiciel de l'OPS en fonction de leur degré de réutilisation.

Les modules/classes ont été répartis en 4 catégories :

- les modules/classes réutilisés,
- les modules/classes réutilisés adaptés,
- les modules/classes réutilisés modifiées,
- les nouveaux modules/classes.

Le terme "adapté" est utilisé pour les classes dont les modifications consistent :

- soit en des mises à jour dues au portage (essentiellement des problèmes de compilation),
- soit à la prise en compte des librairies MLA spécifiques du CGS. Il est à noter que ces librairies ont des interfaces similaires à celles des librairies CSF du PDS-ENVISAT (utilisées pour le SIF).

Le terme "modifié" est utilisé :

- soit lorsque l'implémentation de méthodes évolue,
- soit lorsque la structure de la classe évolue fortement par le rajout et la suppression de méthodes.

Dans le premier cas, nous parlerons de modification d'implémentation tandis que dans le second cas de modification de structure.

### **Stratégie 1 : stratégie de tests unitaires**

#### *Stratégie 1.1 :*

Le principe de la boîte noire est utilisée. Chaque module (classe) est testé en le considérant comme une boîte noire. Un pilote de test permet de simuler l'appel au module (classe). La vérification est réalisée en exploitant les traces générées par chacun des modules (chacune des classes) ainsi que les fichiers produits par chacun des modules (chacune des classes).

Les tests se déroulent de façon incrémentale, c'est-à-dire que l'on commence par les composants ne faisant pas appel à d'autres procédures ou méthodes d'autres composants du logiciel, puis l'on continue par les composants appelant des procédures (méthodes) d'autres modules (classes) de façon à construire progressivement chacun des processus.

Cette stratégie est mise en œuvre :

- pour les nouveaux modules développés,

- pour les modules réutilisés avec modification de structure.

### *Stratégie 1.2 :*

Les classes sont testées lors des tests unitaires de la classe principale (main) de chaque processus.

Cette stratégie est mise en oeuvre dans le cas :

- de modules réutilisés avec de faible modification d'implémentation,
- de modules réutilisés modifiées qui implémentent des traitements linéaires non complexes.

### **Stratégie 2 : \_stratégie de tests d'Intégration/Validation**

Les classes sont testées lors des tests d'intégration et de validation.

Cette stratégie est mise en oeuvre dans le cas :

- des modules réutilisés,
- de modules réutilisés adaptés.

## 8.COMPOSITION DE L'ENSEMBLE

Le composant CMN est constitué d'un ensemble de modules. Chaque module fournit les services liés à une encapsulation ou à une abstraction.

<i>Composant</i>	<i>Classe</i>	<i>Stratégie de test</i>
<b>CMN</b>	CFG__Configuration	2
	CMN__Chaine	2
	CMN__Cmd	1.2
	CMN__CmdTM	1.2
	CMN__Config	2
	CMN__Configuration	2
	CMN__Date	2
	CMN__Fichier	2
	CMN__PGEServices	1.2
	CMN__PGEServicesAPI	2
	CMN__PGEServicesStub	1.2
	CMN__Securite	2
	CMN__StageTM	1.2
	CMN__Trace	2
	CMN__Version	2
	ERR__Exception	2
	JDB__Callback	1.2
	JDB__Client	2
	JDB__Message	2

<i>Composant</i>	<i>Classe</i>	<i>Stratégie de test</i>
	MP__ArretProcess	1.2
	MP__Command	1.2
	MP__Dump	1.2
	MP__HKTMSStatus	1.2
	MP__Spacecraft	1.2
	MP__StageTM	1.2
	MP__UnProcess	2
	SD__Init	2
	TES__Evenement	2

<i>Composant</i>	<i>Classe</i>	<i>Stratégie de test</i>
<b>CMN/DAT</b>	CMN_DAT_Evenement	2
	CMN_DAT_ListeObjets	2
	CMN_DAT_ModeleAction	2
	CMN_DAT_ModeleTache	1.2
	CMN_DAT_ModeleTraitement	1.1
	CMN_DAT_Objct	2
	CMN_DAT_WorkOrder	1.1

<i>Composant</i>	<i>Classe</i>	<i>Stratégie de test</i>
<b>CMN/OBS</b>	CMN_OBS_Observateur	2
	CMN_OBS_Sujet	2

<i>Composant</i>	<i>Classe</i>	<i>Stratégie de test</i>
<b>CMN/PRS</b>	CMN_PRS_DumpFormatTarget	1.1
	CMN_PRS_ErrorReporter	1.1
	CMN_PRS_Parser	1.1

## 9.DESCRPTION DES UNITES DE REALISATION

### 9.1.CLASSE CFG\_\_CONFIGURATION

#### 9.1.1.Spécifications

-----  
Composant : CMN

CLASSE : CFG\_\_Configuration

Commentaire :

Ce module gere la classe CFG\_\_Configuration

-----  
LISTE DES METHODES

-----  
E CFG\_\_Configuration  
E ~CFG\_\_Configuration  
E GetFichier  
E GetRepertoire  
E GetNumerique  
E GetAdresse  
E GetListeAdresse  
E GetListeChaine  
E GetChaine  
E GetValeurEnvironnement

#### 9.1.2.Conception Détaillée

-----  
Methode : CFG\_\_Configuration

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : ~CFG\_\_Configuration

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : GetFichier

Description : Permet de recuperer le nom du fichier dont le nom  
logique est passé en paramètre

Parametres :  
(ENTREE) fichier : nom logique du fichier

Valeur de retour :  
Nom physique du fichier

-----  
Retour de la valeur de la variable d'environnement  
dont le nom passé en paramètre est préfixé par "FIC\_"  
-----

Methode : GetRepertoire

Description : Permet de recuperer le nom du repertoire dont le nom  
logique est passé en paramètre

Parametres :  
(ENTREE) repertoire : nom logique du repertoire

Valeur de retour :  
Nom physique du repertoire

-----  
Retour de la valeur de la variable d'environnement  
dont le nom passé en paramètre est préfixé par "REP\_"  
-----

Methode : GetNumerique

Description : Permet de récupérer la valeur numérique dont le nom  
logique est passé en paramètre

Parametres :  
(ENTREE) numerique : nom logique de la valeur

Valeur de retour :  
Valeur réelle associée, MAXDOUBLE si non connu

-----  
Retour de la valeur de la variable d'environnement  
dont le nom passé en paramètre est préfixé par "NUM\_"  
et transformation de la valeur en double  
-----

Methode : GetAdresse

Description : Permet de récupérer l'adresse dont le nom logique  
est passé en paramètre

Parametres :  
(ENTREE) adresse : nom logique de l'adresse

Valeur de retour :

## Adresse réelle associée

-----  
Retour de la valeur de la variable d'environnement  
dont le nom passé en paramètre est préfixé par "ADR\_"  
-----

Methode : GetListeAdresse

Description : Permet de récupérer la liste d'adresses dont le nom  
logique est passé en paramètre

Parametres :

(ENTREE) adresses : nom logique de la liste d'adresses

Valeur de retour :

Liste d'adresses réelle associée triées

-----  
Récupération de la valeur de la variable d'environnement  
dont le nom passé en paramètre est préfixé par "LISTE\_ADR\_"  
Construction de la liste d'adresses  
Retour de cette dernière  
-----

Methode : GetListeChaine

Description : Permet de récupérer la liste de chaines dont le nom  
logique est passé en paramètre

Parametres :

(ENTREE) chaine : nom logique de la liste de chaines

Valeur de retour :

Liste d'adresses réelle associée

-----  
Récupération de la valeur de la variable d'environnement  
dont le nom passé en paramètre est préfixé par "LISTE\_CHN\_"  
Construction de la liste d'adresses  
Retour de cette dernière  
-----

Methode : GetChaine

Description : Permet de récupérer la liste de chaines dont le nom  
logique est passé en paramètre

Parametres :

(ENTREE) chaine : nom logique de la chaine

Valeur de retour :

Chaine de caractères associée

-----  
Retour de la valeur de la variable d'environnement  
dont le nom passé en paramètre est préfixé par "CHN\_"  
-----

Methode : GetValeurEnvironnement

Description : Permet de récupérer la valeur de la variable



d'environnement dont le nom est passé en paramètre

Parametres :

(ENTREE) variable : nom de la variable d'environnement

Valeur de retour :

Valeur de la variable d'environnement trouvée

-----  
Récupération de la valeur de variable d'environnement  
Transformation de cette dernière en chaîne  
Retour de la valeur

## 9.2.CLASSE CMN\_\_CHAINE

### 9.2.1.Spécifications

-----  
Composant : CMN

CLASSE : CMN\_\_Chaine

Commentaire :

Ce module gère la classe CMN\_\_Chaine, qui permet de manipuler des chaînes de caractères

-----  
LISTE DES METHODES

-----  
E CMN\_\_Chaine  
E ~CMN\_\_Chaine  
E CMN\_\_Chaine  
E CMN\_\_Chaine  
E CMN\_\_Chaine  
E CMN\_\_Chaine  
E CMN\_\_Chaine  
E CMN\_\_Chaine  
E Upper  
E Lower  
E operator long()  
E operator int()  
E operator double()  
E FormaterValeurF  
E FormaterValeurE  
E CopySansLigneComment  
E ExtraireValeurChamp  
E RemplacerValeurPos  
E RemplacerValeurAncienne  
E RemplacerValeurChamp  
E RemplacerValeurChamp  
E RemplacerValeurChamp  
E ExtrairePosAvantChamp  
E ExtrairePosAprsChamp  
E ExtrairePosAvantChamp  
E ExtrairePosAprsChamp  
E VerifierChamp

E RemplacerChaine

## 9.2.2.Conception Détaillée

-----  
Methode : CMN\_\_Chaine

Description : Constructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : ~CMN\_\_Chaine

Description : Destructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : CMN\_\_Chaine

Description : Constructeur avec affectation d'un objet string

Parametres :  
(ENTREE) s : string d'initialisation

Valeur de retour :  
Aucune

-----  
Methode : CMN\_\_Chaine

Description : Constructeur avec affectation d'une chaine de caracteres

Parametres :  
(ENTREE) s : chaine de caracteres d'initialisation

Valeur de retour :  
Aucune

-----  
Methode : CMN\_\_Chaine

Description : Constructeur avec affectation d'un objet CMN\_\_chaine

Parametres :

(ENTREE) chaine : objet CMN\_\_Chaine d'initialisation

Valeur de retour :

Aucune

-----  
Methode : CMN\_\_Chaine

Description : Constructeur avec affectation d'une valeur entiere  
longue

Parametres :

(ENTREE) valeur : valeur entiere longue  
d'initialisation

Valeur de retour :

Aucune

-----  
Methode : CMN\_\_Chaine

Description : Constructeur avec affectation d'une valeur entiere

Parametres :

(ENTREE) valeur : valeur entiere d'initialisation

Valeur de retour :

Aucune

-----  
Methode : CMN\_\_Chaine

Description : Constructeur avec affectation d'une valeur reelle

Parametres :

(ENTREE) valeur : valeur reelle d'initialisation

Valeur de retour :

Aucune

-----  
Methode : Upper

Description : Convertit toute les lettres de la chaine en majuscule

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Dupliquer la chaine

Convertir tous les caractères de la chaine dupliquée en majuscule

Affecter la nouvelle chaine

Libérer la chaine dupliquée  
-----

Methode : Lower

Description : Convertit toute les lettres de la chaine en minuscule

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Dupliquer la chaine

Convertir tous les caractères de la chaine duppliquée en majuscule

Affecter la nouvelle chaine

Liberer la chaine dupliquée  
-----

Methode : operator long()

Description : Convertit la chaine de caractère en valeur entière

Parametres :

(ENTREE) valeur à convertir

Valeur de retour :

long

-----  
Methode : operator int()

Description : Convertit la chaine de caractère en valeur entière

Parametres :

(ENTREE) valeur à convertir

Valeur de retour :

int

-----  
Methode : operator double()

Description : Convertit la chaine de caractère en valeur réelle

Parametres :

Aucun

Valeur de retour :

double

-----  
Methode : FormaterValeurF

Description : Met une valeur au format voulu sous forme de string

Parametres :

(ENTREE) Valeur : Valeur à mettre au format  
FTaille.Precision

(ENTREE) Taille : Taille du string, signe et point  
compris

(ENTREE) Precision : Nombre de chiffres après la virgule

Valeur de retour :

Valeur au format sous forme de string

-----  
Methode : FormaterValeurE

Description : Met une valeur au format voulu sous forme de string

Parametres :

(ENTREE) Valeur : Valeur à mettre au format  
ETaille.Precision

(ENTREE) Taille : Taille du string, signe et point  
compris

(ENTREE) Precision : Nombre de chiffres après la  
virgule

Valeur de retour :

Valeur au format sous forme de string

-----  
Methode : CopySansLigneComment

Description : Recopie les lignes d'un string sans les lignes de  
commentaires

Parametres :

(ENTREE) Original: string original

(ENTREE) Symbole : caractère(s) au début des lignes  
de commentaire

Valeur de retour :

Le string sans les lignes de commentaires

-----  
Methode : ExtraireValeurChamp

Description : Renvoie la valeur d'un champ

Parametres :

(ENTREE) ValeurDest : String où est stockée la valeur

(ENTREE) NomChamp : Nom du champ à rechercher

(ENTREE) Delimiter : Delimiter de la fin du champ

(ENTREE) APartirDe : Position à partir de laquelle ça  
cherche (si rien n'est spécifié,  
ça commence au début du fichier)

Valeur de retour :

npos si le champ n'a pas été trouvé

-----  
Methode : RemplacerValeurPos

Description : Va écrire dans le string source la nouvelle valeur

Parametres :

(ENTREE) Valeur : Valeur à écrire

(ENTREE) Pos : Position où écrire

Valeur de retour :

npos si le champ n'a pas été trouvé

-----  
Methode : RemplacerValeurAncienne

Description : Va écrire dans le string source la nouvelle valeur  
du champ

Parametres :

(ENTREE) ValeurAncienne : Valeur à remplacer

(ENTREE) ValeurNouvelle : Valeur à écrire

Valeur de retour :

npos si le champ n'a pas été trouvé

-----  
Methode : RemplacerValeurChamp

Description : Va écrire dans le string source la nouvelle valeur  
du champ

Parametres :

(ENTREE) Valeur : Valeur à écrire

(ENTREE) NomChamp : Nom du champ à rechercher

Valeur de retour :

npos si le champ n'a pas été trouvé

-----  
Methode : RemplacerValeurChamp

Description : Va écrire dans le string source la nouvelle valeur  
du champ

Parametres :

(ENTREE) Valeur : Valeur à écrire

(ENTREE) NomChamp : Nom du champ à rechercher

(ENTREE) TailleChamp : Nbre de caractère à écrire (met  
des espace en plus si  
nécessaire)

Valeur de retour :

npos si le champ n'a pas été trouvé

-----  
Methode : RemplacerValeurChamp

Description : Va écrire dans le string source la nouvelle valeur  
du champ

Parametres :

(ENTREE) Valeur : Valeur à écrire

(ENTREE) NomChamp : Nom du champ à rechercher

(ENTREE) APartirDe : Position à partir de laquelle

ça cherche

Valeur de retour :  
npos si le champ n'a pas été trouvé

-----  
Methode : ExtrairePosAvantChamp

Description : Renvoie l'adresse pointant sur le champ

Parametres :  
(ENTREE) NomChamp : Nom du champ à rechercher

Valeur de retour :  
Adresse sur le champ, npos si le champ n'est pas  
trouvé

-----  
Methode : ExtrairePosAprsChamp

Description : Renvoie l'adresse pointant juste après le champ

Parametres :  
(ENTREE) NomChamp : Nom du champ à rechercher

Valeur de retour :  
Adresse sur la Valeur du champ, npos si le champ n'a  
pas été trouvé

-----  
Methode : ExtrairePosAvantChamp

Description : Renvoie l'adresse pointant sur le champ

Parametres :  
(ENTREE) NomChamp : Nom du champ à rechercher  
(ENTREE) PosDepart : Endroit à partir duquel ça  
cherche

Valeur de retour :  
Adresse sur le champ, npos si le champ n'est pas  
trouvé

-----  
Methode : ExtrairePosAprsChamp

Description : Renvoie l'adresse pointant juste après le champ

Parametres :  
(ENTREE) NomChamp : Nom du champ à rechercher  
(ENTREE) PosDepart : Endroit à partir duquel ça  
cherche

Valeur de retour :  
Adresse sur la Valeur du champ, npos si le champ  
n'est pas trouvé

-----

Methode : VerifierChamp

Description : Verifie la presence d'un mot-cle

Parametres :

(ENTREE) nomChamp : Nom du champ à rechercher

Valeur de retour :

Aucun. Une exception est émise si le champ  
n'a pas été trouvé.

-----  
Recherche de la position du champ

SI le champ n'a pas été trouvé

émission d'une exception

SINON

vérification que le caractère avant le champ est

un espace

SI ce n'est pas le cas

émission d'une exception

FIN SI

FIN SI

-----  
Methode : RemplacerChaine

Description : Remplace dans une chaîne toutes les occurrences d'une  
sous-chaîne par une autre de même longueur

Parametres :

(ENTREE) AncienneValeur

(ENTREE) NouvelleValeur

Valeur de retour :

Chaîne avec les remplacements effectués

## 9.3.CLASSE CMN\_\_CMD

### 9.3.1.Spécifications

-----  
Composant : CMN

CLASSE : CMN\_\_Cmd

Commentaire :

This class manage the PPF-IASI/PGF interfaces. It centralizes  
the PGF API access

-----  
LISTE DES METHODES

-----  
E SetCommandGroup

E GetCommandGroupStr



## 9.3.2. Conception Détaillée

-----  
Methode : SetCommandGroup

Description : This method stores the command group according to  
the string given as input parameter

Parametres :  
(ENTREE) commandGroup

Valeur de retour :  
t\_CMN\_\_Statut : Statut d'exécution

-----  
TANT QUE command group non trouvé  
Compare type enum to command type given as parameter  
FIN TANT QUE  
Compute status execution according to finding result  
-----

Methode : GetCommandGroupStr

Description : Retourne l'attribut commandGroup

Parametres :  
Aucun

Valeur de retour :  
Command group sous format chaine de caractères  
-----

## 9.4. CLASSE CMN\_\_CMDTM

### 9.4.1. Spécifications

-----  
Composant : CMN  
CLASSE : CMN\_\_CmdTM

Description :  
Cette librairie gère un compte rendu d'exécution d'une commande

-----  
LISTE DES METHODES  
-----

E CMN\_\_CmdTM  
E SetCommandType  
E GetCommandTypeStr

## 9.4.2. Conception Détaillée

Methode : CMN\_\_CmdTM

Description : Constructeur

Parametres :

(ENTREE) commandGroup : Groupe de la commande

Valeur de retour :

Aucune

-----  
Initialiser le type de commande  
Initialiser l'état de la commande  
Initialiser le nom du work order  
-----

Methode : SetCommandType

Description : This method stores the command type according to the  
string given as input parameter

Parametres :

(ENTREE) commandType

Valeur de retour :

t\_CMN\_\_Statut : Statut d'exécution

-----  
TANT QUE command type non trouvé  
    Compare type enum to command type given as parameter  
FIN TANT QUE  
Compute status execution according to finding result  
-----

Methode : GetCommandTypeStr

Description : Retourne l'attribut commandType

Parametres :

Aucun

Valeur de retour :

Command type sous format chaine de caractères

## 9.5. CLASSE CMN\_\_CONFIG

### 9.5.1. Spécifications

-----  
Composant : CMN

CLASSE : CMN\_\_Config

Commentaire :

Ce module gère la classe CMN\_\_Configuration  
en s'appuyant sur le module C CMN\_\_Configuration

-----  
LISTE DES METHODES  
-----

E CMN\_\_Configuration  
E ~CMN\_\_Configuration  
E ChargerFichier  
E ChargerChaine  
E CMN\_\_Configuration\_NombreDeSections  
E CMN\_\_Configuration\_LireSection  
E EcrireValeur  
E LireValeur  
E DetruireSection  
E DetruireChamp  
E SauverChaine  
E SauverFichier  
E AjouterFichier

## 9.5.2.Conception Détaillée

-----

Methode : CMN\_\_Configuration

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : ~CMN\_\_Configuration

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : ChargerFichier

Description : Permet de charger en gestion le contenu d'un  
fichier au format configuration

Parametres :

(ENTREE) Fichier : nom du fichier

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
 CMN\_\_Configuration\_ERR\_MAUVAIS\_ID : Mauvais  
   identifiant de configuration  
 CMN\_\_ERR\_PLUS\_DE\_MEMOIRE : Plus de memoire  
 CMN\_\_ERR\_FICHER\_INTROUVABLE : Fichier introuvable  
 CMN\_\_Configuration\_ERR\_AUCUN\_CHARGEMENT : Aucun  
   élément chargé

-----  
 Methode : ChargerChaine

Description : Permet de charger en gestion le contenu d'une zone  
                   mémoire au format configuration

Parametres :

(ENTREE) chaine

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
 CMN\_\_Configuration\_ERR\_MAUVAIS\_ID : Mauvais  
   identifiant de configuration  
 CMN\_\_ERR\_PLUS\_DE\_MEMOIRE : Plus de memoire  
 CMN\_\_Configuration\_ERR\_AUCUN\_CHARGEMENT : Aucun  
   élément chargé

-----  
 Methode : CMN\_\_Configuration\_NombreDeSections

Description : Permet d'avoir le nombre de sections actuellement  
                   dans la gestion de configuration au format  
                   configuration

Parametres :

(SORTIE) NombreSections : Nombre de sections

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
 CMN\_\_Configuration\_ERR\_MAUVAIS\_ID : Mauvais  
   identifiant de configuration

-----  
 Methode : CMN\_\_Configuration\_LireSection

Description : Permet d'avoir le nom d'une section d'indice i  
                   actuellement dans la gestion de configuration

Parametres :

(ENTREE) Indice  
 (SORTIE) Section : Nom de la section

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
 CMN\_\_ERR\_PLUS\_DE\_MEMOIRE : Plus de memoire  
 CMN\_\_Configuration\_ERR\_MAUVAIS\_ID : Mauvais  
   identifiant de configuration  
 CMN\_\_Configuration\_ERR\_MAUVAIS\_INDICE : Mauvais  
   indice de section

-----  
Methode : EcrireValeur

Description : Permet d'avoir le nom d'une section d'indice i  
actuellement dans la gestion de configuration

Parametres :

(ENTREE) Section  
(ENTREE) Champ  
(ENTREE) Valeur

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
CMN\_\_ERR\_PLUS\_DE\_MEMOIRE : Plus de memoire  
CMN\_\_Configuration\_ERR\_MAUVAIS\_ID : Mauvais  
identifiant de configuration  
CMN\_\_Configuration\_ERR\_MAUVAIS\_INDICE : Mauvais  
indice de section

-----  
Methode : LireValeur

Description : Permet de lire la valeur d'un champ  
dans une configuration en gestion

Parametres :

(ENTREE) Section  
(ENTREE) Champ  
(SORTIE) Valeur

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
CMN\_\_Configuration\_ERR\_MAUVAIS\_ID : Mauvais  
identifiant de configuration  
CMN\_\_ERR\_PLUS\_DE\_MEMOIRE : Plus de memoire  
CMN\_\_Configuration\_ERR\_SECTION\_INEXISTANTE : Section  
inexistante  
CMN\_\_Configuration\_ERR\_CHAMP\_INEXISTANT : Champ  
inexistant

-----  
Methode : DetruireSection

Description : Permet de détruire une section et tous les champs  
associés dans une configuration en gestion

Parametres :

(ENTREE) section : Nom de la section

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
CMN\_\_Configuration\_ERR\_MAUVAIS\_ID : Mauvais  
identifiant de configuration  
CMN\_\_Configuration\_ERR\_SECTION\_INEXISTANTE : Section  
inexistante

Methode : DetruireChamp

Description : Permet de détruire un champ et la section si le champ était le dernier dans une configuration en gestion

Parametres :

(ENTREE) Section : Nom de la section

(ENTREE) Champ : Nom du champ

Valeur de retour :

CMN STATUT OK : Fonctionnement correct

CMN\_Configuration\_ERR\_MAUVAIS\_ID : Mauvais

identifiant de configuration

```
CMN__Configuration_ERR_SECTION_INEXISTANTE : Section
                                             inexistante
```

CMN Configuration ERR CHAMP INEXISTANT : Champ

inexistant

Methode : SauverChaine

Description : Permet de sauvegarder une configuration en gestion dans une zone memoire au format configuration

Parametres :

(ENTREE) chaine

Valeur de retour :

CMN    STATUT OK : Fonctionnement correct

CMN ERR PLUS DE MEMOIRE : Plus de memoire

CMN\_\_Configuration\_ERR\_MAUVAIS\_ID : Mauvais

identifiant de configuration

CMN ERR CREATION FICHER : Erreur de creation de

fichier

Methode : SauverFichier

Description : Permet de sauvegarder une configuration en gestion dans un fichier au format configuration

Parametres :

(ENTREE) Fichier : Fichier pour sauvegarde

Valeur de retour :

CMN STATUT OK : Fonctionnement correct

CMN\_Configuration\_ERR\_MAUVAIS\_ID : Mauvais

identifiant de configuration

CMN\_\_ERR\_CREATION\_FICHER : Erreur de creation de

fichier

Methode : AjouterFichier

Description : Permet de sauvegarder une configuration en gestion dans un fichier au format configuration dans un fichier existant et à la fin

Parametres :

(ENTREE) Fichier : Fichier pour sauvegarde

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct

CMN\_\_Configuration\_ERR\_MAUVAIS\_ID : Mauvais  
identifiant de configuration

CMN\_\_ERR\_CREATION\_FICHIER : Erreur de creation de  
fichier

## 9.6.CLASSE CMN\_\_CONFIGURATION

### 9.6.1.Spécifications

Composant : CMN

MODULE : CMN\_\_Configuration

Commentaire :

Ce module gère les fonctionnalités de gestion de fichier de configuration pour la gestion des fichiers de configuration

Ouvrir permet d'initialiser la gestion

ChargerFichier permet de charger en mémoire le contenu  
d'un fichier de configuration

ChargerChaine permet de charger en mémoire le contenu  
d'une chaine contenant la configuration

NombreDeSections permet d'avoir le nombre de sections de  
la conf

LireSection permet de lire la valeur d'une section

EcrireValeur permet d'ecrire une valeur d'un champ dans  
une section

LireValeur permet de lire la valeur d'un champ dans une  
section

DetruireSection permet de détruire une section avec tous  
ses champs

DetruireValeur permet de détruire un champ et la section  
associée si s'etait le dernier champ

SauverFichier permet de sauver le contenu de la mémoire  
de configuration dans un fichier

AjouterFichier permet de sauver le contenu de la mémoire  
de configuration dans un fichier a la fin  
d'un fichier

SauverChaine permet de sauver le contenu de la mémoire  
de configuration dans une chaine

Fermer permet de terminer la gestion de configuration et  
libérer

#### LISTE DES PROCEDURES

E CMN\_\_Configuration\_Ouvrir

E CMN\_\_Configuration\_ChargerFichier

```

E CMN__Configuration_ChargerChaine
E CMN__Configuration_NombreDeSections
E CMN__Configuration_LireSection
E CMN__Configuration_EcrireValeur
E CMN__Configuration_LireValeur
E CMN__Configuration_DetruireSection
E CMN__Configuration_DetruireChamp
E CMN__Configuration_SauverChaine
E CMN__Configuration_SauverFichier
E CMN__Configuration_AjouterFichier
E CMN__Configuration_Fermer
I CMN__Configuration_ControllerId
I CMN__Configuration_RechercherSection
I CMN__Configuration_RechercherIndiceSection
I CMN__Configuration_RechercherChamp
I CMN__Configuration_AllouerSection
I CMN__Configuration_AllouerChamp
I CMN__Configuration_AffecterValeur
I CMN__Configuration_Interpreter
I CMN__Configuration_ChcherCaractere
I CMN__Configuration_CalculerLongueurSection
I CMN__Configuration_CopierChaineSection
I CMN__Configuration_AllerApresCr
I CMN__Configuration_CalculerLongueurChamp
I CMN__Configuration_CopierChaineChamp
I CMN__Configuration_AllerApresEgal
I CMN__Configuration_CalculerLongueurValeur
I CMN__Configuration_CopierChaineValeur
I CMN__Configuration_CalculerLongueurConfiguration
I CMN__Configuration_SoustraireBlancEnFin
I CMN__Configuration_DumpTrace
I CMN__Configuration_SauverFichierMode
I CMN__Configuration_TestVide
I CMN__Configuration_SansBlancDevantDerriere
I CMN__Configuration_FusionnerSections
I CMN__Configuration_CompareSections
I CMN__Configuration_TransfertSection

```

## 9.6.2. Conception Détaillée

-----

Procedure : CMN\_\_Configuration\_Ouvrir

Description : Initialise une configuration en rendant un  
identifiant.  
Cet identifiant sera utilisé sur toutes les  
procédures

Parametres :

(SORTIE) IdConfiguration : Identifiant de  
configuration

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct



CMN\_\_ERR\_PLUS\_DE\_MEMOIRE : Plus de mémoire

-----  
Allouer une structure de gestion  
SI l'allocation est en echec  
    sortir en erreur  
FIN SI  
Initialiser la structure  
Positionner l'identifiant en retour  
-----

Procédure : CMN\_\_Configuration\_ChargerFichier

Description : Permet de charger en gestion le contenu d'un  
fichier au format configuration

Parametres :

(ENTREE) IdConfiguration : Identifiant de  
configuration  
(ENTREE) fichier : Nom du fichier

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
CMN\_\_CONFIGURATION\_ERR\_MAUVAIS\_ID : Mauvais  
identifiant de configuration  
CMN\_\_ERR\_PLUS\_DE\_MEMOIRE : Plus de memoire  
CMN\_\_ERR\_FICHER\_INTROUVABLE : Fichier introuvable  
CMN\_\_CONFIGURATION\_ERR\_AUCUN\_CHARGEMENT : Aucun  
élément chargé

-----  
Controler l'identifiant  
Si mauvais identifiant sortir en erreur  
Ouvrir le fichier  
S'il y a un probleme a l'ouverture  
    sortir en erreur  
Lire la taille du fichier  
Si erreur sur la lecture d'information fichier sortir en erreur  
Recuperer la taille du fichier  
Si le fichier est vide sortir en erreur  
Allouer une zone memoire de la taille du fichier  
Si plus de memoire sortir en erreur  
Charger le fichier dans la memoire  
Appeler l'interpreteur sur la zone memoire  
Fusionner les sections rajoutées si nécessaire si il  
y a doublon  
Fermer le fichier  
Libérer la mémoire fichier  
Trace en debug  
-----

Procédure : CMN\_\_Configuration\_ChargerChaine

Description : Permet de charger en gestion le contenu d'une zone  
memoire au format configuration

Parametres :

(ENTREE) IdConfiguration : Identifiant de  
configuration  
(ENTREE) chaine : Pointeur de memoire

(ENTREE) longueur : Longueur de la memoire

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
CMN\_\_CONFIGURATION\_ERR\_MAUVAIS\_ID : Mauvais  
identifiant de configuration  
CMN\_\_ERR\_PLUS\_DE\_MEMOIRE : Plus de memoire  
CMN\_\_CONFIGURATION\_ERR\_AUCUN\_CHARGEMENT : Aucun  
élément chargé

-----  
Controler l'identifiant  
Si mauvais identifiant sortir en erreur  
Appeler l'interpreteur sur la zone memoire  
Fusionner les sections rajoutées si nécessaire si il y a doublon  
Trace en debug  
-----

Procédure : CMN\_\_Configuration\_NombreDeSections

Description : Permet d'avoir le nombre de sections actuellement  
dans la gestion de configuration au format  
configuration

Parametres :

(ENTREE) IdConfiguration : Identifiant de  
configuration  
(SORTIE) nombre : Nombre de sections

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
CMN\_\_CONFIGURATION\_ERR\_MAUVAIS\_ID : Mauvais  
identifiant de configuration

-----  
Controler l'identifiant  
Si mauvais identifiant sortir en erreur  
Sortir avec le nombre  
-----

Procédure : CMN\_\_Configuration\_LireSection

Description : Permet d'avoir le nom d'une section d'indice i  
actuellement dans la gestion de configuration

Parametres :

(ENTREE) IdConfiguration : Identifiant de  
configuration  
(ENTREE) indice : Indice (0 -> n)  
(SORTIE) valeur : Nom de la section

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
CMN\_\_ERR\_PLUS\_DE\_MEMOIRE : Plus de memoire  
CMN\_\_CONFIGURATION\_ERR\_MAUVAIS\_ID : Mauvais  
identifiant de configuration  
CMN\_\_CONFIGURATION\_ERR\_MAUVAIS\_INDICE : Mauvais  
indice de section  
-----

```
Controler l'identifiant
Si mauvais identifiant sortir en erreur
Si indice hors bornes sortir en erreur
Se positionner sur la section
Allouer une taille pour le retour
SI plus de mémoire positionner le code erreur
SINON
    Affecter le nom de la section
    Positionner le code retour OK
FIN SI
Sortir avec le compte-rendu adapte
```

-----

Procédure : CMN\_\_Configuration\_EcrireValeur

Description : Permet d'écrire la valeur d'un champ dans une section dans une configuration en gestion.  
Si le champ n'existe pas il est créé, sinon il est mis à jour.  
Si la section n'existe pas, elle est créée

Parametres :

(ENTREE) IdConfiguration : Identifiant de configuration  
(ENTREE) section : Nom de la section  
(ENTREE) champ : Nom du champ  
(ENTREE) valeur : valeur du champ

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
CMN\_\_ERR\_PLUS\_DE\_MEMOIRE : Plus de mémoire  
CMN\_\_CONFIGURATION\_ERR\_MAUVAIS\_ID : Mauvais identifiant de configuration  
CMN\_\_ERR\_PARAMETRE\_NULL : Section ou champ null ou vide

-----

```
Controler l'identifiant
Si mauvais identifiant sortir en erreur
Rechercher la section avec le nom donné en paramètre
SI il n'y a pas de section avec ce nom
    Creer la section
    Si problème de mémoire sortir en erreur
FIN SI
Allouer une memoire pour la valeur
Si probleme de memoire sortir en erreur
Affecter la valeur
Rechercher le champ dans la section avec le nom donné en paramètre
SI il n'y a pas de champ avec ce nom
    Creer le champ
    Si problème de mémoire sortir en erreur
SINON le champ existe deja
    Libérer l'ancienne valeur
FIN SI
Affecter le (nouveau) pointeur
Trace en debug
```

-----

Procédure : CMN\_\_Configuration\_LireValeur

Description : Permet de lire la valeur d'un champ dans une configuration en gestion

Parametres :

(ENTREE) IdConfiguration : Identifiant de configuration  
(ENTREE) section : Nom de la section  
(ENTREE) champ : Nom du champ  
(SORTIE) valeur : valeur du champ

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
CMN\_\_CONFIGURATION\_ERR\_MAUVAIS\_ID : Mauvais identifiant de configuration  
CMN\_\_ERR\_PLUS\_DE\_MEMOIRE : Plus de memoire  
CMN\_\_CONFIGURATION\_ERR\_SECTION\_INEXISTANTE : Section inexistante  
CMN\_\_CONFIGURATION\_ERR\_CHAMP\_INEXISTANT : Champ inexistant

-----  
Controler l'identifiant

Si mauvais identifiant sortir en erreur

Rechercher la section avec le nom donne en parametre

S'il n'y a pas de section avec ce nom

rendre section inexistante

Rechercher le champ dans la section avec le nom donné en paramètre

S'il n'y a pas de champ avec ce nom

rendre champ inexistant

Allouer une taille pour le retour

SI plus de memoire positionner le code erreur

SINON copier la valeur

FIN SI  
-----

Procédure : CMN\_\_Configuration\_DetruireSection

Description : Permet de detruire une section et tous les champs associes dans une configuration en gestion

Parametres :

(ENTREE) IdConfiguration : Identifiant de configuration  
(ENTREE) section : Nom de la section

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
CMN\_\_CONFIGURATION\_ERR\_MAUVAIS\_ID : Mauvais identifiant de configuration  
CMN\_\_CONFIGURATION\_ERR\_SECTION\_INEXISTANTE : Section inexistante  
-----

Controler l'identifiant

Si mauvais identifiant sortir en erreur

S'il n'y a pas de section avec ce nom

rendre section inexistante

Rechercher la section avec le nom donne en parametre  
S'il n'y a pas de section  
    rendre section inexistante  
Retailler la table  
Libérer la chaine champs  
Libérer la section  
Actualiser le nombre de sections  
Trace en debug

-----  
Procédure : CMN\_\_Configuration\_DetruireChamp

Description : Permet de detruire un champ et la section si le  
                  champ était le dernier dans une configuration en  
                  gestion

Parametres :

(ENTREE) IdConfiguration : Identifiant de  
  configuration  
(ENTREE) section : Nom de la section  
(ENTREE) champ : Nom du champ

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
CMN\_\_CONFIGURATION\_ERR\_MAUVAIS\_ID : Mauvais  
  identifiant de configuration  
CMN\_\_CONFIGURATION\_ERR\_SECTION\_INEXISTANTE : Section  
  inexistante  
CMN\_\_CONFIGURATION\_ERR\_CHAMP\_INEXISTANT : Champ  
  inexistant

-----  
Controler l'identifiant  
Si mauvais identifiant sortir en erreur  
Rechercher la section avec le nom donne en parametre  
S'il n'y a pas de section avec ce nom  
    rendre section inexistante  
Positionner le code retour a section inexistante par default  
TANT QUE il y a une section faire  
    SI l'élément courant a le meme nom que le champ en paramètre  
        refaire les chainages des champ  
        libérer le champ  
        si c'était le dernier, libérer la section  
        positionner le code retour  
    FIN SI  
FIN TANT QUE  
Trace en debug

-----  
Procédure : CMN\_\_Configuration\_SauverChaine

Description : Permet de sauvegarder une configuration en gestion  
                  dans une zone memoire au format configuration

Parametres :

(ENTREE) IdConfiguration : Identifiant de  
  configuration  
(ENTREE/SORTIE) chaine : Pointeur zone memoire pour  
  allocation  
(ENTREE/SORTIE) longueur : Longueur de la zone

mémoire

Valeur de retour :

```
CMN__STATUT_OK : Fonctionnement correct
CMN__ERR_PLUS_DE_MEMOIRE : Plus de memoire
CMN__CONFIGURATION_ERR_MAUVAIS_ID : Mauvais
                                identifiant de configuration
CMN__ERR_CREATION_FICHIER : Erreur de creation de
                                fichier
```

```

Controler l'identifiant
Si mauvais identifiant sortir en erreur
SI il y a des choses a sauver
    Allouer une zone memoire pour le retour
    SI plus de mémoire sortir en erreur
    SINON
        POUR toutes les sections
            sauver le nom de section
            POUR tous les champs
                sauver le nom de champ et la valeur
            FIN POUR
        FIN POUR
    FIN SI
Allouer une zone memoire pour le retour
FIN SI

```

Procedure : CMN Configuration SauverFichier

Description : Permet de sauvegarder une configuration en gestion dans un fichier au format configuration

Parametres :

```
(ENTREE) IdConfiguration : Identifiant de
                             configuration
(ENTREE) fichier : Fichier pour sauvegarde
```

Valeur de retour :

```
CMN__STATUT_OK : Fonctionnement correct
CMN__CONFIGURATION_ERR_MAUVAIS_ID : Mauvais
                                identifiant de configuration
CMN__ERR_CREATION_FICHER : Erreur de creation de
                                fichier
```

```
Procedure : CMN__Configuration_AjouterFichier
```

Description : Permet de sauvegarder une configuration en gestion dans un fichier au format configuration en ajout sur un fichier existant

Parametres :

```
(ENTREE) IdConfiguration : Identifiant de
                             configuration
(ENTREE) fichier : Fichier pour sauvegarde
```

Valeur de retour :

CMN STATUT OK : Fonctionnement correct

CMN\_\_CONFIGURATION\_ERR\_MAUVAIS\_ID : Mauvais  
identifiant de configuration  
CMN\_\_ERR\_CREATION\_FICHER : Erreur de creation de  
fichier

-----  
Procédure : CMN\_\_Configuration\_Fermer

Description : Permet de liberer une configuration en gestion.

Parametres :

(ENTREE) IdConfiguration : Identifiant de  
configuration

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
CMN\_\_CONFIGURATION\_ERR\_MAUVAIS\_ID : Mauvais  
identifiant de configuration

-----  
Controler l'identifiant  
Si mauvais identifiant sortir en erreur  
Recuperer le nombre de section  
POUR toutes les sections en commençant par la fin faire  
libérer la chaine champs  
libérer la section  
FIN POUR

-----  
Procédure : CMN\_\_Configuration\_ControlerId

Description : Controle la validite de l'identifiant passé et rend  
un pointeur sur la structure de gestion associee

Parametres :

(ENTREE) IdConfiguration : Identifiant de  
configuration

Valeur de retour :

Pointeur sur la struture de gestion associée à  
l'identifiant  
NULL : Mauvais identifiant

-----  
SI le champs self pointé par l'identifiant n'est pas égal à  
l'identifiant  
sortir en erreur  
SINON rendre la structure de gestion de configuration  
FIN SI

-----  
Procédure : CMN\_\_Configuration\_RechercherSection

Description : Permet de rechercher une section dans la  
configuration en cours et rendre le pointeur sur elle  
si elle existe

Parametres :

(ENTREE) IdConf : Pointeur sur la configuration  
(ENTREE) section : Nom de la section

Valeur de retour :

Pointeur sur la section

NULL : Section introuvable

-----  
S'il n'y a pas de section

rendre NULL et sortir

Sortir avec le pointeur  
-----

Procédure : CMN\_\_Configuration\_RechercherIndiceSection

Description : Permet de rechercher un indice de section dans la configuration en cours

Parametres :

(ENTREE) IdConf : Pointeur sur la configuration

(ENTREE) section : Nom de la section

Valeur de retour :

>=0 : indice de section

-1 : non trouve

-----  
S'il n'y a pas de section en parametre

rendre NULL et sortir

Si le nombre de section est égal à 0

rendre NULL et sortir

Regarder si la derniere section accedee n'est pas celle demandee

SI oui rendre l'indice

rendre NULL et sortir

SINON parcourir à partir du dernier acces jusqu'à la fin

FIN SI

SI pas trouve jusqu'a la fin

sinon parcourir a partir du premier jusqu'au dernier acces

FIN SI

S'il n'y a pas de section

rendre NULL et sortir

Sortir avec l'indice  
-----

Procédure : CMN\_\_Configuration\_RechercherChamp

Description : Permet de rechercher un champ dans la configuration en cours et rendre le pointeur sur elle si elle existe

Parametres :

(ENTREE) section : Pointeur sur la section

(ENTREE) champ : Nom du champ

Valeur de retour :

Pointeur sur le champ

NULL : Champ introuvable

-----  
S'il n'y a pas de champ

rendre NULL et sortir

TANT QUE qu'il y a un champ faire

si l'élément courant a le même nom que le champ en parametre



```
rendre le pointeur dessus
FIN TANT QUE
Sortir avec NULL
```

-----

Procédure : CMN\_\_Configuration\_AllouerSection

Description : Permet de creer une section dans la configuration  
en cours et rendre le pointeur sur elle

Parametres :

(ENTREE) IdConf : Pointeur sur la configuration  
(ENTREE) section : Nom de la section

Valeur de retour :

Pointeur sur la section  
NULL : Plus de mémoire

-----

```
Allouer une mémoire pour la structure de gestion
Si problème d'allocation
    sortir avec NULL
Allouer une mémoire pour le nom de la section
Si probleme d'allocation
    sortir avec NULL
Copier le parametre dans la memoire allouee
Affecter le pointeur dans la structure
Rendre le pointeur
```

-----

Procédure : CMN\_\_Configuration\_AllouerChamp

Description : Permet de creer un champ dans la configuration  
en cours et rendre le pointeur sur elle

Parametres :

(ENTREE) section : Pointeur sur la section  
(ENTREE) champ : Nom du champ

Valeur de retour :

Pointeur sur le champ  
NULL : Plus de memoire

-----

```
Allouer une mémoire pour la structure de gestion
Si problème d'allocation
    sortir avec NULL
Allouer une mémoire pour le nom du champ
Si problème d'allocation
    sortir avec NULL
Copier le paramètre dans la mémoire allouée
Affecter le pointeur dans la structure
SI c'est le premier champ
    le mettre dans le pointeur de tete de section
SINON
    aller sur le dernier champ
    raccrocher à la queue du dernier
FIN SI
Rendre le pointeur
```

-----

Procédure : CMN\_\_Configuration\_AffecterValeur

Description : Permet de mettre à jour la valeur d'un champ  
si il y avait déjà une valeur elle est libérée

Paramètres :

(ENTREE) champ : Pointeur sur le champ  
(ENTREE) valeur : Valeur

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
CMN\_\_ERR\_PLUS\_DE\_MEMOIRE : Plus de mémoire

-----  
Allouer une mémoire pour la valeur  
Si problème d'allocation  
sortir avec NULL  
Copier le paramètre dans la mémoire allouée  
Si il y a une valeur dans le champ  
on la libère  
Positionner la nouvelle valeur pour le champ  
-----

Procédure : CMN\_\_Configuration\_Interpreter

Description : Permet de créer les sections et les champs  
pour la gestion de configuration en analysant  
un buffer mémoire au format configuration

Paramètres :

(ENTREE) IdConf  
(ENTREE) mémoire : Buffer  
(ENTREE) taille\_memoire : Taille du buffer

Valeur de retour :

CMN\_\_STATUT\_OK : Fonctionnement correct  
CMN\_\_ERR\_PARAMETRE\_NULL : Un paramètre nul  
CMN\_\_ERR\_PLUS\_DE\_MEMOIRE : Plus de mémoire  
CMN\_\_CONFIGURATION\_ERR\_AUCUN\_CHARGEMENT : Aucun  
élément chargé

-----  
Contrôler les paramètres  
sortie en erreur si un paramètre est NULL  
Initialiser une boucle sur la mémoire  
TANT QUE on est pas à la fin de la mémoire faire  
SELON le caractère  
CAS '[' : début de section  
Calculer la longueur du nom de section  
SI le calcul est correct  
Allouer une zone mémoire pour stocker le nom de  
la section  
SI l'allocation est correcte  
Copier le nom dans la mémoire  
SI la section n'existe pas  
Créer un élément section dans la  
configuration  
Si la création se passe mal  
Positionner le flag de fin de boucle et  
-----

```
le compte-rendu PLUS_DE_MEMOIRE
FIN SI
FIN SI
Libérer la zone memoire
Aller a la fin de la ligne
SINON positionner le flag de fin de boucle et
le compte-rendu PLUS_DE_MEMOIRE
FIN SI
FIN SI
CAS '\r': fin de ligne
CAS '\n': fin de ligne
CAS EOF: fin buffer
CAS '#': commentaire
Default: il s'agit d'un champ
Calculer la longueur du nom de champ
SI le calcul est correct
Allouer une zone mémoire pour stocker le nom
du champ
SI l'allocation est correcte
Copier le nom dans la mémoire
Aller sur le =
Si le calcul est correct
Allouer une zone memoire pour stocker la
valeur du champ
Si l'allocation est correcte
Copier la valeur dans la memoire
Créer un champ avec une valeur
Si la section n'existe pas
Créer un élément champ dans la
configuration
SI la création se passe mal
Positionner le flag de fin de
boucle et le compte-rendu
PLUS_DE_MEMOIRE
SINON
Mettre la valeur dans le champ
Si erreur d'allocation
mémoire positionner le
flag de fin de boucle
et le compte-rendu
PLUS_DE_MEMOIRE
FIN SI
SINON
Mettre à jour la valeur dans
le champ
Si erreur d'allocation mémoire
positionner le flag de fin de
boucle et le compte-rendu
PLUS_DE_MEMOIRE
FIN SI
Libérer la zone mémoire
SINON positionner le flag de fin de boucle
et le compte-rendu PLUS_DE_MEMOIRE
FIN SI
Libérer la zone memoire
Aller a la fin de la ligne
SINON positionner le flag de fin de boucle et
le compte-rendu PLUS_DE_MEMOIRE
```

```
FIN SI
FIN TANT QUE
SI le statut est OK
    SI aucune création de sections ou de champs n'a été faite
        Positionner une erreur de "pas de creation"
    FIN SI
FIN SI
```

-----

Procédure : CMN\_\_Configuration\_ChcherCaractere

Description : Permet de se positionner sur un caractère  
significatif du buffer, cad autre que ESPACE et TAB

Parametres :

(ENTREE/SORTIE) pt\_courant : Pointeur sur le début  
de traitement  
(ENTREE) pt\_fin : Pointeur sur la fin à ne pas  
dépasser

Valeur de retour :

Le caractère significatif ou EOF en fin de buffer

-----

```
Au début tant que le caractère est ESPACE ou TAB et qu'on
n'est pas à la fin, se déplacer dans le buffer
SI le pointeur de déplacement n'est pas à la fin
    positionner le caractère de retour et mettre
    à jour la valeur du pointeur courant
SINON le retour est EOF
FIN SI
```

-----

Procédure : CMN\_\_Configuration\_CalculerLongueurSection

Description : Permet de calculer le nombre de caractères du nom  
d'une section dans le buffer, le pointeur courant  
doit être positionner sur le [ ou juste apres

Parametres :

(ENTREE) pt\_courant : Pointeur sur le début  
de traitement  
(ENTREE) pt\_fin : Pointeur sur la fin à ne pas  
dépasser

Valeur de retour :

Le nombre de caractère ou 0

-----

```
Si le pointeur courant est sur le [ et pas en fin,
sauter le caractère
Au debut tant que le caractère est ESPACE ou TAB et qu'on
n'est pas à la fin, se déplacer dans le buffer
Si le pointeur de deplacement n'est pas a la fin
    compter les caractères jusqu'a la fin, ou un ESPACE
    ou un TAB ou un CR ou un ]
```

-----

Procédure : CMN\_\_Configuration\_CopierChaineSection

Description : Permet de copier le nom d'une section situe

dans le buffer dans une chaine passée en paramètre,  
le pointeur courant doit etre positionner sur le [  
ou juste apres.  
La chaine passée en paramètre doit avoir la taille  
rendue par  
CMN\_\_Configuration\_CalculerLongueurSection + 1  
pour le NULL

## Parametres :

(ENTREE) pt\_courant : Pointeur sur le début  
de traitement  
(ENTREE) pt\_fin : Pointeur sur la fin à ne pas  
dépasser  
(ENTREE/SORTIE) chaine : Pointeur sur la chaine  
pour copie  
(ENTREE) lg : Longueur à copier

## Valeur de retour :

Aucune

-----  
Si le pointeur courant est sur le [ et pas en fin,  
sauter le caractère  
Tant que le caractère est ESPACE ou TAB et qu'on n'est  
pas à la fin, se déplacer dans le buffer  
Si le pointeur de déplacement n'est pas à la fin  
copier lg caractères  
-----

## Procédure : CMN\_\_Configuration\_AllerAprèsCr

Description : Permet de se positionner sur un caractère après le  
prochain CR

## Parametres :

(ENTREE/SORTIE) pt\_courant : Pointeur sur le début  
de traitement  
(ENTREE) pt\_fin : Pointeur sur la fin à ne pas  
dépasser

## Valeur de retour :

Aucune

-----  
Tant que le caractère n'est pas CR et qu'on n'est pas à la fin,  
se déplacer dans le buffer  
Si le pointeur de déplacement n'est pas à la fin  
passer après le cr  
Mettre à jour le pointeur en paramètre  
-----

## Procédure : CMN\_\_Configuration\_CalculerLongueurChamp

Description : Permet de calculer le nombre de caractères du nom  
d'un champ dans le buffer

## Parametres :

(ENTREE) pt\_courant : Pointeur sur le début  
de traitement  
(ENTREE) pt\_fin : Pointeur sur la fin à ne pas

dépasser

Valeur de retour :

Le nombre de caractère ou 0

-----  
Purge les blancs au début

Tant que le caractère est ESPACE ou TAB et qu'on n'est pas à la fin, se déplacer dans le buffer

SI le pointeur de déplacement n'est pas à la fin

compter les caractères jusqu'à la fin, ou un ESPACE ou

un TAB ou un CR ou un =

si on n'a pas trouvé de =

fin anormale on se positionne à la fin

FIN SI

-----  
Procédure : CMN\_\_Configuration\_CopierChaineChamp

Description : Permet de copier le nom d'une section située dans le buffer dans une chaîne passée en paramètre.

La chaîne passée en paramètre doit avoir la taille rendue par CMN\_\_Configuration\_CalculerLongueurChamp + 1 pour le NULL

Paramètres :

(ENTREE) pt\_courant : Pointeur sur le début de traitement

(ENTREE) pt\_fin : Pointeur sur la fin à ne pas dépasser

(ENTREE/SORTIE) chaine : Pointeur sur la chaîne pour copie

(ENTREE) lg : Longueur à copier

Valeur de retour :

Aucune

-----  
Purge des blancs au début

Tant que le caractère est ESPACE ou TAB et qu'on n'est pas à la fin, se déplacer dans le buffer

SI le pointeur de déplacement n'est pas à la fin

copier lg caractères

FIN SI

-----  
Procédure : CMN\_\_Configuration\_AllerAprèsEgal

Description : Permet de se positionner sur un caractère après le prochain =

Paramètres :

(ENTREE/SORTIE) pt\_courant : Pointeur sur le début de traitement

(ENTREE) pt\_fin : Pointeur sur la fin à ne pas dépasser

Valeur de retour :

Aucune

-----  
Tant que le caractère n'est pas CR et pas = et qu'on n'est pas à la fin, se déplacer dans le buffer  
Si le pointeur de déplacement n'est pas à la fin et n'est pas CR passer après le =  
Mettre à jour le pointeur en paramètre  
-----

Procédure : CMN\_\_Configuration\_CalculerLongueurValeur

Description : Permet de calculer le nombre de caractères de la valeur d'un champ dans le buffer, le pointeur courant doit être positionner sur le = ou juste après

Parametres :

(ENTREE) pt\_courant : Pointeur sur le début de traitement  
(ENTREE) pt\_fin : Pointeur sur la fin à ne pas dépasser

Valeur de retour :

Le nombre de caractère ou 0

-----  
Si le pointeur courant est sur le = et pas en fin, sauter le caractère  
Au début tant que le caractère est ESPACE ou TAB et qu'on n'est pas à la fin, se déplacer dans le buffer  
Si le pointeur de déplacement n'est pas à la fin compter les caractères jusqu'à la fin ou un CR  
-----

Procédure : CMN\_\_Configuration\_CopierChaineValeur

Description : Permet de copier la valeur d'un champ située dans le buffer dans une chaîne passée en paramètre, le pointeur courant doit être positionner sur le = ou juste après. La chaîne passée en paramètre doit avoir la taille rendue par CMN\_\_Configuration\_CalculerLongueurValeur + 1 pour le NULL

Parametres :

(ENTREE) pt\_courant : Pointeur sur le début de traitement  
(ENTREE) pt\_fin : Pointeur sur la fin à ne pas dépasser  
(ENTREE/SORTIE) chaine : Pointeur sur la chaîne pour copie  
(ENTREE) lg : Longueur à copier

Valeur de retour :

Aucune

-----  
Si le pointeur courant est sur le [ et pas en fin, sauter le caractère  
Au début tant que le caractère est ESPACE ou TAB et qu'on n'est pas à la fin, se déplacer dans le buffer

Si le pointeur de déplacement n'est pas à la fin  
copier lg caractères

-----  
Procédure : CMN\_\_Configuration\_CalculerLongueurConfiguration

Description : Permet de d'avoir la longueur (chaîne ascii) de la  
configuration en gestion

Paramètres :

(ENTREE) IdConf

Valeur de retour :

Longueur de la configuration

-----  
Procédure : CMN\_\_Configuration\_SoustraireBlancEnFin

Description : Permet de d'avoir la longueur significatif d'un  
champ en enlevant les blancs en fin

Paramètres :

(ENTREE) pt\_debut : Pointeur sur le début  
significatif

(ENTREE/SORTIE) lg : Longueur avec les blancs  
éventuels

Valeur de retour :

Aucune

-----  
Si la longueur n'est pas positive sortir  
TANT QUE il y a des blancs à la fin diminuer la longueur  
FIN TANT QUE

-----  
Procédure : CMN\_\_Configuration\_DumpTrace

Description : Permet de tracer l'arborescence des sections et  
champs

Paramètres :

(ENTREE) IdConf : Pointeur sur la configuration

Valeur de retour :

Aucune

-----  
Procédure : CMN\_\_Configuration\_SauverFichierMode

Description : Permet de sauvegarder une configuration en gestion  
dans un fichier au format configuration en  
exploitant le mode d'écriture du fichier passé  
en paramètre

Paramètres :

(ENTREE) IdConfiguration : Identifiant de  
configuration

(ENTREE) fichier : Fichier pour sauvegarde

(ENTREE) mode : Mode de sauvegarde



Valeur de retour :

CMN STATUT OK : Fonctionnement correct

CMN CONFIGURATION ERR MAUVAIS ID : Mauvais

identifiant de configuration

CMN\_\_ERR\_CREATION\_FICHER : Erreur de creation de  
fichier

```

    Contrôler l'identifiant
Si mauvais identifiant sortir en erreur
Ouvrir le fichier
S'il y a un problème à l'ouverture
    sortir en erreur
SI pas de section ne rien faire
SINON
    POUR toutes les sections
        sauver le nom de section
        POUR tous les champs
            sauver le nom de champ et la valeur
        FIN POUR
    FIN POUR
FIN SI
Fermer le fichier

```

Procedure : CMN\_\_Configuration\_TestVide

Description : Permet de tester si un champ est vide

Parametres :

(ENTREE) valeur : Valeur a tester

Valeur de retour :

CMN STATUT OK : Valeur non vide

CMN\_\_STATUT\_NOK : Valeur vide

Procedure : CMN Configuration SansBlancDevantDerriere

Description : Permet de supprimer les blancs devant et derriere une valeur

Parametres :

(ENTREE) in : Valeur à transformer

(SORTIE) out : Valeur transformée

Valeur de retour :

Aucune

```
Procedure : CMN  Configuration FusionnerSections
```

Description : Permet de sauvegarder une configuration en gestion dans un fichier au format configuration en exploitant le mode d'écriture du fichier passé en parametre

Parametres :

(ENTREE) IdConf : Identifiant de configuration

Valeur de retour :

Aucune

-----  
Dupliquer la table des sections

Trier le tableau

Eliminer les doublons  
-----

Procédure : CMN\_\_Configuration\_CompareSections

Description : Permet de comparer deux pointeurs de section

Parametres :

(ENTREE) sec1 : Pointeur de pointeur section 1

(ENTREE) sec2 : Pointeur de pointeur section 2

Valeur de retour :

>0 : sec 1 > sec2

<0 : sec 1 < sec2

0 : sec 1 = sec 2  
-----

Procédure : CMN\_\_Configuration\_TransfertSection

Description : Permet de mettre à jour une section avec une autre

Parametres :

(ENTREE) IdConf : Identifiant configuration

(ENTREE) pt\_section1 : Pointeur section 1

(ENTREE) pt\_section2 : Pointeur section 2

Valeur de retour :

Aucune  
-----

Si il n'y a pas de champs dans section2 sortir

Positionner le dernier accès sur la section 1

Transférer les champs de section 2 dans section 1

Positionner le dernier accès sur la section 2

Détruire la section

## 9.7.CLASSE CMN\_\_DATE

### 9.7.1.Spécifications

-----  
Composant : CMN

CLASSE : CMN\_\_Date

Commentaire :

Ce module gère la classe CMN\_\_Date. Il permet de faire des calculs et des comparaisons sur les dates.

Il manipule des date comprises entre le 1er Janvier

1970 et le 5 Février 2036.

Référence jour Julien standard = 1 Janvier 4712 Avant JC 12:00:00

Référence jour Julien CNES = 1 janvier 1950 Après JC 00:00:00

-----  
LISTE DES METHODES  
-----

E CMN\_\_Heure  
E CMN\_\_Heure  
E CMN\_\_Heure  
E ~CMN\_\_Heure  
E Chaîne  
E Chaîne  
E Chaîne  
E Formater  
E operator =  
E operator +  
E operator -  
E operator ==  
E operator >=  
E operator <=  
I TesterValidite  
I ConvertirHeureEnDuree  
I ConvertirDureeEnHeure  
I CMN\_\_Jour  
I CMN\_\_Jour  
I CMN\_\_Jour  
I Chaîne  
I ~CMN\_\_Jour  
I NomMois  
I Chaîne  
I Chaîne  
I Formater  
I operator =  
I operator ==  
I operator >=  
I operator <=  
I TesterValidite  
I NormaliserDate  
I ConvertirJourEnTemps  
I ConvertirTempsEnJour  
I CMN\_\_Date  
I CMN\_\_Date  
I CMN\_\_Date  
I CMN\_\_Date  
I ~CMN\_\_Date  
I DateCourante  
I DateCourantePrecise  
I LaDateCourantePrecise  
I DateInit  
I JourJulien  
I JourJulienCnes  
I JourJulienCnes  
I JourJulien  
I Formater  
I Chaîne  
I Chaîne

I Chaîne  
I operator =  
I operator +  
I operator -  
I operator +  
I operator -  
I operator -  
I operator ==  
I operator >=  
I operator <=  
I ConvertirDateEnTemps  
I ConvertirTempsEnDate  
I AjouterDuree

## 9.7.2.Conception Détaillée

-----  
Methode : CMN\_\_Heure

Description : Constructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : CMN\_\_Heure

Description : Constructeur

Parametres :  
(ENTREE) sHeure : Heure sous forme de chaine

Valeur de retour :  
Aucune

-----  
Methode : CMN\_\_Heure

Description : Constructeur

Parametres :  
(ENTREE) sHeure : Chaine à partir de laquelle l'heure  
doit être construite  
(ENTREE) sFormat : Format permettant d'analyser la  
chaine

Valeur de retour :  
Aucune

-----  
Methode : ~CMN\_\_Heure

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : Chaine

Description : Convertit une heure en chaine de caractère

Parametres :

Aucun

Valeur de retour :

t\_Chaine : Heure résultat sous forme de chaine

-----  
Methode : Chaine

Description : Positionne la valeur d'une heure à partir d'une chaine de caractère

Parametres :

(ENTREE) uneHeure : Heure sous forme de chaine

Valeur de retour :

CMN\_\_STATUT\_OK : format de la chaine valide

CMN\_\_STATUT\_NOK : format de la chaine invalide

-----  
Methode : Chaine

Description : Affecte la valeur de l'heure à partir d'une chaine et d'un format

Parametres :

(ENTREE) sHeure : Heure sous forme de chaine

(ENTREE) sFormat : Format de la date

Valeur de retour :

CMN\_\_STATUT\_OK : format de la chaine valide

CMN\_\_STATUT\_NOK : format de la chaine invalide

-----  
SI le format ne spécifie pas les fractions de secondes

    parser la chaine en entrée selon le format

    mettre à jour la forme interne

SINON le format spécifie les fractions de secondes

    Séparer le format en 3 parties:

        . Avant %s

        . %s format pour décoder la partie fractionnaire des secondes

        . Apres %s

    Parser la chaine en entrée selon le format Préfixe

    Récupérer les fractions de secondes

Construire Format Reel  
Parser la chaine en entrée selon le format Réel  
Mettre à jour la forme interne

FIN SI

Mettre à jour le status en fonction de la validité

-----  
Methode : Formater

Description : Convertit une heure en chaine de caractère suivant  
le format.  
Le format est celui utilisé par la fonction strftime

Parametres :

(ENTREE) Format : format de l'heure  
(ENTREE) NbDigitFrac : Nombre de digit des fractions  
de secondes

Valeur de retour :

La date formatée ou "" si erreur

-----  
SI le format ne spécifie pas les fractions de secondes  
SINON le format spécifie les fractions de secondes  
Formater les fractions de secondes  
Remplacer le %s dans le format par les fractions de seconde  
Formater avec le nouveau format  
FIN SI

-----  
Methode : operator =

Description : Affectation d'une heure

Parametres :

(ENTREE) uneHeure : Heure à affecter

Valeur de retour :

CMN\_\_Heure : Heure résultat

-----  
Methode : operator +

Description : Ajoute une heure

Parametres :

(ENTREE) uneHeure : Heure à ajouter

Valeur de retour :

CMN\_\_Heure : Heure résultat

-----  
Methode : operator -

Description : Soustrait une heure

Parametres :

(ENTREE) uneHeure : Heure à soustraire

Valeur de retour :

CMN\_\_Heure : Heure résultat

-----  
Methode : operator ==

Description : Teste l'égalité de deux heures

Parametres :

(ENTREE) unJour : Jour de comparaison

Valeur de retour :

bool : Résultat du test (true si égalité)

-----  
Methode : operator >=

Description : Teste si l'heure est supérieure ou égale à l'heure  
passée en paramètre

Parametres :

(ENTREE) uneHeure : Heure de comparaison

Valeur de retour :

bool : Résultat du test

-----  
Methode : operator <=

Description : Teste si l'heure est inférieure ou égale à l'heure  
passée en paramètre

Parametres :

(ENTREE) uneHeure : Heure de comparaison

Valeur de retour :

bool : Résultat du test

-----  
Methode : TesterValidite

Description : Teste la validité d'une heure

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : ConvertirHeureEnDuree

Description : Convertit une heure en Duree

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : ConvertirDureeEnHeure

Description : Convertit la Duree en heure

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : CMN\_\_Jour

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : CMN\_\_Jour

Description : Constructeur à partir d'une chaine

Parametres :

(ENTREE) sJour : jour sous forme de chaine

Valeur de retour :

Aucune

-----  
Methode : CMN\_\_Jour

Description : Constructeur à partir d'une chaine

Parametres :

(ENTREE) sJour : jour sous forme de chaine

(ENTREE) sFormat : format de la chaine

(ENTREE) sLangue : langue à utiliser

Valeur de retour :

Aucune

-----  
Methode : Chaine

Description : Affecte la valeur du jour à partir d'une chaine et d'un format

Parametres :

(ENTREE) sJour : Jour sous forme de chaine

(ENTREE) sFormat : Format du jour

(ENTREE) sLangue : Langue pour le décodage de la date



Valeur de retour :

CMN\_\_STATUT\_OK : format de la chaine valide

CMN\_\_STATUT\_NOK : format de la chaine invalide

-----  
Mémorisation du LcTime courant  
Positionnement du Nouveau LcTime  
Parser la chaine en entrée selon le format  
Mettre à jour la forme interne  
Mettre à jour le statut de retour  
Restauration de l'ancien LcTime  
-----

Methode : ~CMN\_\_Jour

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : NomMois

Description : Extrait le mois de la date

Parametres :

Aucun

Valeur de retour :

Le mois de la date

-----  
Methode : Chaine

Description : Retourne la date sous forme de chaine

Parametres :

Aucun

Valeur de retour :

Date sous forme de chaine (format jj/mm/aaaa)

-----  
Methode : Chaine

Description : Affecte la valeur de la date à partir d'une chaine

Parametres :

(ENTREE) sDate : Date sous forme de chaine (format  
jj-mmm-aaaa)

Valeur de retour :

CMN\_\_STATUT\_OK : format de la chaine valide

CMN\_\_STATUT\_NOK : format de la chaine invalide  
-----

Methode : Formater

Description : Convertit un jour en chaine de caractère suivant le format.  
Le format est celui utilisé par la fonction strftime

Parametres :

(ENTREE) Format : format du jour  
(ENTREE) sLangue : Langue

Valeur de retour :

La date formatée ou "" si erreur

-----  
Mémorisation du LcTime courant  
Positionnement du Nouveau LcTime  
Conversion  
Restauration de l'ancien LcTime  
-----

Methode : operator =

Description : Operation d'affection entre deux jours

Parametres :

(ENTREE) unJour : Jour origine

Valeur de retour :

CMN\_\_Jour& : Jour destination

-----  
Methode : operator ==

Description : Teste l'égalité de deux jours

Parametres :

(ENTREE) unJour : Jour de comparaison

Valeur de retour :

bool : Résultat du test

-----  
Methode : operator >=

Description : Teste si le Jour est supérieur ou égal au Jour passé en paramètre

Parametres :

(ENTREE) unJour : Jour de comparaison

Valeur de retour :

bool : Résultat du test

-----  
Methode : operator <=

Description : Teste si le Jour est inférieur ou égal au Jour passé en paramètre

Parametres :

(ENTREE) unJour : Jour de comparaison

Valeur de retour :

bool : Résultat du test

-----  
Methode : TesterValidite

Description : Verifie q'un jour est cohérent

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Normalisation

Si la date n'a pas bougé après normalisation, c'est qu'elle  
était cohérente

-----  
Methode : NormaliserDate

Description : Normalise une date

Le jour doit être supérieur au 01/01/1970

Parametres :

Aucun

Valeur de retour :

CMN\_\_STATUT\_OK : le jour est valide

CMN\_\_STATUT\_NOK : le jour est invalide

-----  
Convertit la date en une date normalisee

Prend en compte la zone de temps locale

-----  
Methode : ConvertirJourEnTemps

Description : Convertit un jour en temps

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Verifier si la date est postérieure au 1er Janvier 70

-----  
Methode : ConvertirTempsEnJour

Description : Convertit le temps en un jour

Parametres :

Aucun

Valeur de retour :  
Aucune

-----  
Methode : CMN\_\_Date

Description : Constructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : CMN\_\_Date

Description : Constructeur de recopie

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : CMN\_\_Date

Description : Constructeur

Parametres :  
(ENTREE) sDate : Chaine à partir de laquelle la date  
doit être construite

Valeur de retour :  
Aucune

-----  
Methode : CMN\_\_Date

Description : Constructeur

Parametres :  
(ENTREE) sDate : Chaine à partir de laquelle la date  
doit être construite  
(ENTREE) sFormat : Format permettant d'analyser la  
chaine  
(ENTREE) sLangue : Langue selon laquelle la chaine  
est analysée

Valeur de retour :  
Aucune

-----  
Methode : ~CMN\_\_Date

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : DateCourante

Description : Positionne la date à la date du Jour

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : DateCourantePrecise

Description : Récupère la date courante en microsecondes

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Récupérer la date courante en millisecondes  
-----

Methode : LaDateCourantePrecise

Description : Fournit la date courante en microsecondes

Parametres :

(ENTREE) precision : Précision du champ tv\_usec (0  
à 6)

(SORTIE) laDate : La date courante au format timeval

Valeur de retour :

Aucune

-----  
Methode : DateInit

Description : Initialise la date au 1er janvier 2000

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : JourJulien

Description : Retourne le Jour Julien

Nombre de jours depuis le 1er Janvier 4712 Avant JC  
à 12h 0m 0s

Parametres :

Aucun

Valeur de retour :

Nombre de jours julien CNES

-----  
Methode : JourJulienCnes

Description : Retourne le Jour Julien CNES

Nombre de jours depuis le 1er Janvier 1950 à 0h 0m 0s

Parametres :

Aucun

Valeur de retour :

Nombre de jours julien CNES

-----  
Methode : JourJulienCnes

Description : Construire la date à partir du nombre de jours  
Julien Cnes

Parametres :

(ENTREE) JourJulienCnes : Nombre de jours julien CNES

Valeur de retour :

Aucune

-----  
Methode : JourJulien

Description : Construire la date à partir du nombre de jours Julien

Parametres :

(ENTREE) JourJulien : Nombre de jours

Valeur de retour :

Aucune

-----  
Convertir le jour julien en temps UTC  
Mettre à jour la représentation interne  
-----

Methode : Formater

Description : Convertit une date en chaine de caractère suivant  
le format.  
Le format est celui utilisé par la fonction strftime

Parametres :

(ENTREE) Format : format de la date  
(ENTREE) NbDigitFrac : format de la date  
(ENTREE) sLangue : format de la date

Valeur de retour :  
La date formatée ou "" si erreur

-----  
Mémorisation du LcTime courant  
Positionnement du Nouveau LcTime  
SI le format ne spécifie pas les fractions de secondes  
SINON le format spécifie les fractions de secondes  
Formater les fractions de secondes  
Remplacer le %s dans le format par les fractions de seconde  
Formater avec le nouveau format  
FIN SI  
Restauration de l'ancien LcTime  
-----

Methode : Chaine

Description : Retourne la date sous forme de chaine

Parametres :  
Aucun

Valeur de retour :  
Date sous forme de chaine

-----  
Methode : Chaine

Description : Affecte la valeur de la date à partir d'une chaine

Parametres :  
(ENTREE) sDate : Date sous forme de chaine

Valeur de retour :  
CMN\_\_STATUT\_OK : format de la chaine valide  
CMN\_\_STATUT\_NOK : format de la chaine invalide  
-----

Methode : Chaine

Description : Affecte la valeur de la date à partir d'une chaine  
et d'un format

Parametres :  
(ENTREE) sDate : Date sous forme de chaine  
(ENTREE) sFormat : Format de la date  
(ENTREE) sLangue : Langue pour le décodage de la date

Valeur de retour :  
CMN\_\_STATUT\_OK : format de la chaine valide  
CMN\_\_STATUT\_NOK : format de la chaine invalide  
-----

Mémorisation du LcTime courant  
Positionnement du Nouveau LcTime  
SI le format ne spécifie pas les fractions de secondes  
Parser la chaine en entrée selon le format  
Mettre à jour la forme interne

SINON le format spécifie les fractions de secondes  
Séparer le format en 3 parties:  
  . Avant %s  
  . %s format pour décoder la partie fractionnaire des  
  secondes  
  . Après %s  
Parser la chaine en entrée selon le format Préfixe  
On recupere tous les digits des fractions de secondes  
(même les 0 au debut)  
  Construire Format Reel  
  Parser la chaine en entrée selon le format Réel  
  Mettre à jour la forme interne  
FIN SI  
Mettre à jour la validité de la date et le statut de retour  
Restauration de l'ancien LcTime

-----  
Methode : operator =

Description : Opération d'affectation d'une date

Parametres :

(ENTREE) uneDate : Date origine

Valeur de retour :

CMN\_\_Date& : Date destination

-----  
Methode : operator +

Description : Ajoute une durée

Parametres :

(ENTREE) Duree : Durée à ajouter

Valeur de retour :

CMN\_\_Date : Résultat de l'opération

-----  
Methode : operator -

Description : Soustrait une durée

Parametres :

(ENTREE) Duree : Durée à soustraire

Valeur de retour :

CMN\_\_Date : Résultat de l'opération

-----  
Methode : operator +

Description : Ajoute une durée exprimée en heure

Parametres :

(ENTREE) uneHeure : Heure à ajouter

Valeur de retour :

CMN\_\_Date : Résultat de l'opération



-----  
Methode : operator -

Description : Soustrait une duree exprimée en heure

Parametres :  
          (ENTREE) uneHeure : Heure à soustraire

Valeur de retour :  
          CMN\_\_Date : Résultat de l'opération

-----  
Methode : operator -

Description : Soustrait une date à une autre date.  
          Le résultat est une durée en seconde

Parametres :  
          (ENTREE) uneHeure : Heure à soustraire

Valeur de retour :  
          double : Résultat de l'opération

-----  
Methode : operator ==

Description : Teste l'égalité de deux dates

Parametres :  
          (ENTREE) uneDate : Date de comparaison

Valeur de retour :  
          bool : Résultat du test

-----  
Methode : operator >=

Description : Teste si la date est supérieure ou égale à la date  
          passée en paramètre

Parametres :  
          (ENTREE) uneDate : Date de comparaison

Valeur de retour :  
          bool : Résultat du test

-----  
Methode : operator <=

Description : Teste si la date est inférieure ou égale à la date  
          passée en paramètre

Parametres :  
          (ENTREE) uneDate : Date de comparaison

Valeur de retour :  
          bool : Résultat du test

-----  
Methode : ConvertirDateEnTemps

Description : Convertit une date en temps

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : ConvertirTempsEnDate

Description : Convertit le temps en une date

Parametres :

(ENTREE) Temps : Temps en secondes

Valeur de retour :

Aucune

-----  
Methode : AjouterDuree

Description : Ajoute une durée

Parametres :

(ENTREE) Duree : Durée à ajouter

Valeur de retour :

Résultat de l'opération

## 9.8.CLASSE CMN\_\_FICHIER

### 9.8.1.Spécifications

-----  
Composant : CMN

MODULE : CMN\_\_Fichier

Commentaire :

Ce module propose un ensemble de services concernant les fichiers :

- TrouverFichier: retourne la liste des fichiers correspondant à une expression régulière
- TrouverFichierEx: retourne la liste des infos sur les fichiers correspondant à une expression régulière. Elle permet de spécifier une fonction de sélection ainsi qu'une fonction de tri
- CopierFichier: copie un fichier dans un autre fichier

-----  
LISTE DES PROCEDURES  
-----

E CMN\_\_RenommerFichier  
E CMN\_\_TrouverFichiers  
E CMN\_\_TrouverFichiersEx  
E CMN\_\_Copier  
E CMN\_\_OrdreAlphaFichier  
E CMN\_\_PlusVieuxQue  
E CMN\_\_PlusRecentQue  
E CMN\_\_TriDateCroissante  
E CMN\_\_CreerArborescence  
E CMN\_\_RecuperEtatFileSystem  
E CMN\_\_Acces

## 9.8.2.Conception Détaillée

  
-----

Procédure : CMN\_\_RenommerFichier

Description : Déplace un fichier d'un répertoire à un autre

Parametres :

(ENTREE) source : Fichier à déplacer  
(ENTREE) dest : Fichier destination

Valeur de retour :

0 si le déplacement a été effectué  
-1 si erreur

-----  
Détermination d'un fichier temporaire  
Copie du fichier dans le fichier temporaire  
Renommage du fichier temporaire avec le nom du fichier cible  
Suppression du fichier source  
-----

Procédure : CMN\_\_TrouverFichiers

Description : Rend la liste des fichiers répondant à un pattern  
expression reguliere en local avec la date des  
fichiers

Parametres :

(ENTREE) path: Path des fichier en local  
(ENTREE) pattern : Expression régulière décrivant  
le fichier  
(SORTIE) liste\_fichiers : Liste des fichiers

Valeur de retour :

CMN\_\_STATUT\_OK si la fonction a réussi  
CMN\_\_STATUT\_NOK si la fonction a échoué  
CMN\_\_ERR\_PERMISSION\_REFUSEE : Problème d'accès au  
répertoire

CMN\_\_ERR\_PLUS\_DE\_MEMOIRE : Plus de mémoire  
CMN\_\_ERR\_MAUVAIS\_FORMAT si le pattern est incorrect  
CMN\_\_ERR\_FICHER\_INTROUVABLE : Pas de commande

-----  
Vider la liste de retour  
Rechercher les fichiers qui match le pattern  
Recopier les noms des fichiers dans la liste de retour  
Retourner le statut  
-----

Procédure : CMN\_\_TrouverFichiersEx

Description : Rend la liste des fichiers qui match un pattern.  
Si une fonction de sélection est passée en argument  
alors une 2eme selection est pratiquée selon ce qui  
est défini par cette fonction.  
Si une fonction de tri est passée en argument alors  
la liste est triée selon cette fonction.  
La fonction de selection est appelée (Si elle existe)  
pour chaque fichier qui match le pattern avec en  
argument :  
    . un paramètre libre qui est celui passée en  
      argument de CMN\_\_TrouverFichiersEx,  
    . le nom du fichier  
    . la structure stat associé à ce fichier  
Le fichier pour lequel cette fonction est appelée,  
sera inclu dans la liste des fichiers si la fonction  
de sélection retourne true.  
La fonction de tri est appelée pour comparer 2  
fichiers de la liste.  
Elle doit retourner :  
    < 0 si fic1 < fic2  
    = 0 si fic1 == fic2  
    > 0 si fic1 > fic2

Parametres :

(ENTREE) path: Path des fichier en local  
(ENTREE) pattern : Expression régulière décrivant  
                  le fichier  
(SORTIE) liste\_fichiers : Liste des fichiers  
(ENTREE) fnSelect : Fonction de sélection  
(ENTREE) fnTri : Fonction de comparaison utilisée  
                  pour le tri  
(ENTREE) param : Paramètre libre pour l'utilisateur

Valeur de retour :

CMN\_\_STATUT\_OK si la fonction a réussi  
CMN\_\_STATUT\_NOK si la fonction a échoué  
CMN\_\_ERR\_PERMISSION\_REFUSEE: Problème d'accès au  
                                répertoire  
CMN\_\_ERR\_PLUS\_DE\_MEMOIRE : Plus de memoire  
CMN\_\_ERR\_MAUVAIS\_FORMAT si le pattern est incorrect  
CMN\_\_ERR\_FICHER\_INTROUVABLE : Pas de commande

-----  
Vider la liste des infos  
Compilation de l'expression régulière  
Allouer un buffer pour la lecture

```
TANT QUE on a rien trouvé et que le nombre max de tentatives
n'est pas dépassé
    pour tous les fichiers du repertoire
        comparer le nom au pattern (expression reguliere)
        SI le fichier correspond au critere de selection
            Insérer le fichier dans la liste des
            fichiers sélectionnés
        FIN SI
    Fermer le répertoire
    SI une prochaine tentative est autorisée et si on
    a rien trouvé
        Attendre
    FIN SI
FIN TANT QUE boucle sur le nombre de tentatives
Libérer l'expression régulière compilée
SI il faut trier
    Trier
FIN SI
```

-----  
Procédure : CMN\_\_Copier

Description : Copie fichierSrc dans fichierDest

Parametres :

```
(ENTREE) path : Path des fichier en local
(ENTREE) pattern : Expression régulière décrivant
                  le fichier
(SORTIE) liste_fichiers : Liste des fichiers
```

Valeur de retour :

```
CMN__STATUT_OK si la fonction a réussi
CMN__STATUT_NOK si la fonction a échoué
```

-----  
SI le fichier Destination existe et si une erreur doit être  
retournée en cas d'existence  
 Retourner une erreur

FIN SI

Ouverture en lecture du fichier source et en écriture du  
fichier destination

SI ouvertures réussies

```
    Copier le fichier source dans le fichier Dest par bloc
    de 2048 caractères
```

```
    Lecture d'un buffer dans le fichier source
```

```
    Ecriture du buffer dans le fichier dest
```

```
    Fermeture des 2 fichiers
```

SINON erreur d'ouverture d'au moins un des fichiers

FIN SI

-----  
Procédure : CMN\_\_OrdreAlphaFichier

Description : Compare 2 objets CMN\_\_InfoFichier selon un ordre  
alphabétique sur les noms de fichier

Parametres :

```
(ENTREE) CMN__InfoFichier info1
(ENTREE) CMN__InfoFichier info2
```

Valeur de retour :

Retourne 0 si info1 == info2  
Retourne <0 si info1 < info2  
Retourne >0 si info1 > info2

-----  
Procédure : CMN\_\_PlusVieuxQue

Description : Compare la date de création d'un fichier à une date donnée

Parametres :

(ENTREE) void \* ParamDate  
(ENTREE) t\_chaine NomDuFichier  
(ENTREE) CMN\_\_Stat Stat

Valeur de retour :

Retourne true si le fichier a été créé avant la date passée en paramètre

-----  
Procédure : CMN\_\_PlusRecentQue

Description : Compare la date de création d'un fichier à une date donnée

Parametres :

(ENTREE) void \* ParamDate  
(ENTREE) t\_chaine NomDuFichier  
(ENTREE) CMN\_\_Stat Stat

Valeur de retour :

Retourne true si le fichier a été créé après la date passée en paramètre

-----  
Procédure : CMN\_\_TriDateCroissante

Description : Compare 2 objets CMN\_\_InfoFichier selon les dates de création des fichiers

Parametres :

(ENTREE) CMN\_\_InfoFichier info1  
(ENTREE) CMN\_\_InfoFichier info2

Valeur de retour :

Retourne 0 si info1 == info2  
Retourne <0 si info1 < info2  
Retourne >0 si info1 > info2

-----  
Procédure : CMN\_\_CreerArborescence

Description : Crée si nécessaire les répertoires utiles à l'existence du chemin d'accès passé en paramètre

Parametres :

(ENTREE) Chemin  
(ENTREE) droits

Valeur de retour :  
Aucune

-----  
Crée un chemin tel que il y ait un slash à la fin  
Positionne Pos après le premier Slash  
Positionne Pos avant le second Slash  
Crée le chemin partiel si il n'existe pas  
Positionne Pos après le slash  
Positionne Pos avant le slash suivant  
-----

Procédure : CMN\_\_RecuperEtatFileSystem

Description : Retourne le nombre d'octets utilisés et libres pour  
le filesystem dont le chemin est passé en paramètre

Paramètres :

(ENTREE) t\_Chaine Chemin  
(SORTIE) double Taille  
(SORTIE) double Libre

Valeur de retour :  
Aucune

-----  
Procédure : CMN\_\_Acces

Description : Teste si les droits d'un fichier de donnée sont  
corrects (rw-r--r--). Le chemin d'accès est passé  
en paramètre

Paramètres :

(ENTREE) Chemin  
(ENTREE) Droits

Valeur de retour :  
True si les droits rw-r--r-- sont positionnés  
False sinon  
-----

## 9.9.CLASSE CMN\_\_PGESERVICES

### 9.9.1.Spécifications

-----  
Composant : CMN

CLASSE : CMN\_\_PGEServices

Commentaire :

Ce fichier contient la classe CMN\_\_PGEServices, qui gère les  
interfaces de type PPF-IASI/MCS.  
Il regroupe les accès à l'API MCS

-----  
LISTE DES METHODES  
-----

E CMN\_\_PGEServices  
E ~CMN\_\_PGEServices  
E MLAINitialisation  
E MLATerminate  
E MLASendFunctionStatus  
E MLASendEvent  
E GetPGFCommand  
E SendPGFCommandTM  
E SendPGFStageTM

## 9.9.2.Conception Détaillée

  
-----

Methode : CMN\_\_PGEServices

Description : Constructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
The constructor retrieves the current installation type in order  
to initialize the PGF API : nominal or stub  
Initialisation du spacecraft par default à NONE  
-----

Methode : ~CMN\_\_PGEServices

Description : Destructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : MLAINitialisation

Description : This method initialises the interface between an OPS  
processus and the PGF services

Parametres :  
Aucun

Valeur de retour :  
t\_CMN\_\_PGEStatus : Statut d'exécution



-----  
Methode : MLATerminate

Description : This method closes the interface between an OPS  
processus and the PGF services

Parametres :  
Aucun

Valeur de retour :  
t\_CMN\_\_PGESatus : Statut d'exécution

-----  
Methode : MLASendFunctionStatus

Description : This method manages MLA sendFunction Status services

Parametres :  
(ENTREE) HKTM

Valeur de retour :  
t\_CMN\_\_PGESatus : Statut d'exécution

-----  
Recupérer la date Courante  
Extraire de l'objet HKTMStatus les informations a envoyer au MCS  
Si la valeur de la HKTM envoyée est de type integer alors l'encoder  
Initialiser la valeur du paramètre Validity: toujours à TRUE  
Appeler le services PGF avec les données fabriquées

-----  
Methode : MLASendEvent

Description : This method call the appropriate object in order to  
send a log or a trace event message.

Parametres :  
(ENTREE) message : Message à envoyer

Valeur de retour :  
t\_CMN\_\_Statut : Statut d'exécution

-----  
Recupérer la date du message  
Recupérer le libellé  
Construire le message Name  
SI le message est de type Log ALORS  
Compléter le nom du message  
Construire le message a l'aide des informations recus  
Envoyer le message de log  
SINON (type TRACE)  
Compléter le nom du message  
Construire le message a l'aide des informations recus  
Envoyer le message de trace  
FIN SI

-----  
Methode : GetPGFCommand

Description : This method call the appropriate object in order to

retreive a new command.

Parametres :

(SORTIE) command : Commande reçue

Valeur de retour :

t\_CMN\_\_Statut : Statut d'exécution

```
-----
Récupération de la commande présente sur le stdin
SI la commande est valide
    Construire une commande
    Initialisation d'une instance de commande à l'aide des
    informations extraites
    Rechercher le type de la commande passée en parametre
FIN SI
-----
```

Methode : SendPGFCommandTM

Description : This method call the appropriate method in order to  
send a command TM message.

Parametres :

(ENTREE) commandTM : Message commandTM

Valeur de retour :

t\_CMN\_\_Statut : Statut d'exécution

```
-----
Construire le message Command TM a envoyer
Envoyer le message commandTM au PGF
-----
```

Methode : SendPGFStageTM

Description : This method call the appropriate method in order to  
send a stage TM message.

Parametres :

(ENTREE) stageTM : Message stageTM

Valeur de retour :

t\_CMN\_\_Statut : Statut d'exécution

```
-----
Construire le message Stage a envoyer
Envoyer le message StageTM au PGF
-----
```

## 9.10.CLASSE CMN\_\_PGESERVICESAPI

### 9.10.1.Spécifications

Composant : CMN

CLASSE : CMN\_\_PGEServicesAPI

Commentaire :

Cette classe implémente dans le cas du fonctionnement en  
STANDALONE l'accès aux API du MCS et PGF

-----  
LISTE DES METHODES  
-----

E CMN\_\_PGEServicesAPI  
E ~CMN\_\_PGEServicesAPI  
E MLASInitialisation  
E MLATerminate  
E MLASendFunctionStatus  
E MLALogEvent  
E MLALogTrace  
E GetPGFCommand  
E SendPGFCommandTM  
E SendPGFStageTM

## 9.10.2.Conception Détaillée

-----

Methode : CMN\_\_PGEServicesAPI

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Initialiser les paramètres de scrutation du stdin :  
- masque de socket en lecture sur le stdin (fd = 0)  
- le timeout a 0  
-----

Methode : ~CMN\_\_PGEServicesAPI

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : MLASInitialisation

Description : This method manages MLA library initialisation  
function call

Parametres :

(ENTREE) facilityName

(ENTREE) MPFlag : boolean indicates if the process  
which call the services is the MP  
process or not

Returned Status :

t\_CMN\_\_PGESStatus : Statut d'exécution

-----  
Methode : MLATerminate

Description : This method manages MLA librairy terminaison  
function call

Parametres :

Aucun

Valeur de retour :

t\_CMN\_\_PGESStatus : Statut d'exécution

-----  
Methode : MLASendFunctionStatus

Description : This method manages MLA snedFunction Status services

Parametres :

(ENTREE) timeEvt : Temps OS courant

(ENTREE) HKTMparam

Valeur de retour :

t\_CMN\_\_PGESStatus : Statut d'exécution

-----  
Methode : MLALogEvent

Description : This method manages MLA librairy log event function  
call

Parameters :

(ENTREE) timeEvt : Temps OS courant

(ENTREE) event : eventName + eventText

Valeur de retour :

t\_CMN\_\_PGESStatus : Statut d'exécution

-----  
Methode : MLALogTrace

Description : This method manages MLA librairy log trace function  
call

Parameters :

(ENTREE) timeTrace : Temps OS courant

(ENTREE) trace : traceName + traceText

Valeur de retour :

t\_CMN\_\_PGESStatus : Statut d'exécution

-----  
Methode : GetPGFCommand

Description : This method extracts the available command from the  
stdin queue

Parameters :

(SORTIE) command : Commande récupérée  
(SORTIE) validCmd

Valeur de retour :

t\_CMN\_\_PGStatus : Statut d'exécution  
-----

RAZ des masques

Copie du masque (il est modifié par select)

Appel de select

SELON retour du select

CAS retour = -1: erreur systeme exception SOCKET\_ERROR

CAS retour = 0: timeout

default: on récupère la chaine de caractères sur le stdin

FIN SELON  
-----

Methode : SendPGFCommandTM

Description : This method provides the command TM to the PGF  
from the stdout queue

Parametres :

(ENTREE) commandTM : Command TM

Valeur de retour :

t\_CMN\_\_PGStatus : Statut d'exécution  
-----

Envoi de la commande TM  
-----

Methode : SendPGFStageTM

Description : This method provides the stage TM to the PGF from  
the stdout queue

Parametres :

(ENTREE) stageTM : StageTM

Valeur de retour :

t\_CMN\_\_PGStatus : Statut d'exécution  
-----

Envoi de la stage TM

## 9.11.CLASSE CMN\_\_PGESERVICESSTUB

### 9.11.1.Spécifications

-----  
Composant : CMN

CLASSE : CMN\_\_PGEServicesStub

Commentaire :

Cette classe implémente dans le cas du fonctionnement en  
STANDALONE l'accès aux API du MCS et PGF

-----  
LISTE DES METHODES

-----  
E CMN\_\_PGEServicesStub  
E ~CMN\_\_PGEServicesStub  
E MLAInitialisation  
E MLATerminate  
E MLASendFunctionStatus  
E MLALogEvent  
E MLALogTrace  
E GetPGFCommand  
E SendPGFCommandTM  
E SendPGFStageTM

### 9.11.2.Conception Détaillée

-----  
Methode : CMN\_\_PGEServicesStub

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Recupérer la date courante pour concaténer avec les noms de  
fichier

Ouvrir les fichiers dédiés stockage :

- des log events
- des traces
- des cmdTM et stage TM
- des HKTM status

-----  
Methode : ~CMN\_\_PGEServicesStub

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Fermer les fichiers dédiés stockage :

- des log events
  - des traces
  - des cmdTM et stage TM
  - des HKTM status
- 

Methode : MLAMInitialisation

Description : This method manages MLA library initialisation  
function call

Parametres :

- (ENTREE) facilityName
- (ENTREE) MPFlag : boolean indicates if the process  
which call the services is the MP  
process or not.

Valeur de retour :

t\_CMN\_\_PGStatus : Statut d'exécution

-----  
Methode : MLATerminate

Description : This method manages MLA library terminaison function  
call

Parametres :

Aucun

Valeur de retour :

t\_CMN\_\_PGStatus : Statut d'exécution

-----  
Methode : MLASendFunctionStatus

Description : This method manages MLA snedFunction Status services

Parametres :

- (ENTREE) timeEvt : Temps OS courant
- (ENTREE) HKTMparam

Valeur de retour :

t\_CMN\_\_PGStatus : Statut d'exécution

-----  
Formattage de la date en string

Ecrire le statut dans le fichier trace dédié

-----  
Methode : MLALogEvent

Description : This method manages MLA library log event function  
call

Parametres :

(ENTREE) timeEvt : Temps OS courant  
(ENTREE) event : eventName + eventText

Valeur de retour :

t\_CMN\_\_PGESStatus : Statut d'exécution

-----  
Formattage de la date en string  
Ecrire le status dans le fichier trace dédiée  
-----

Methode : MLALogTrace

Description : This method manages MLA librairy log trace function  
call

Parametres :

(ENTREE) timeTrace : Temps OS courant  
(ENTREE) trace : traceName + traceText

Valeur de retour :

t\_CMN\_\_PGESStatus : Statut d'exécution

-----  
Formattage de la date en string  
Ecrire le status dans le fichier trace dédiée  
-----

Methode : GetPGFCommand

Description : This method extracts the available command from the  
stdin queue

Parameters :

(SORTIE) command : Commande récupérée  
(SORTIE) validCmd

Valeur de retour :

t\_CMN\_\_PGESStatus : Statut d'exécution

-----  
Récupérer dans le répertoire de commandes l'ensemble des  
commandes disponibles  
Extraire la lière commande de la liste  
Ouvrir le fichier de commande et en extraire la commande  
Fermer et détruire le fichier de commande  
-----

Methode : SendPGFCommandTM

Description : This method provides the command TM to the PGF  
from the stdout queue

Parametres :

(ENTREE) commandTM : Command TM

Valeur de retour :

t\_CMN\_\_PGESStatus : Statut d'exécution



-----  
Formattage de la date courante en string  
Ecrire le statut dans le fichier trace dédié  
-----

Methode : SendPGFStageTM

Description : This method provides the stage TM to the PGF from  
the stdout queue

Parametres :

(ENTREE) stageTM : StageTM

Valeur de retour :

t\_CMN\_\_PGStatus : Statut d'exécution  
-----

Formattage de la date courante en string  
Ecrire le statut dans le fichier trace dédié

## 9.12.CLASSE CMN\_\_SECURITE

### 9.12.1.Spécifications

-----  
Composant : CMN

MODULE : CMN\_\_Securite

Commentaire :

Ce module gère les fonctionnalités de gestion des aspects  
sécurité:

Inhiber les cores: CMN\_\_Securite\_NoCore

Capter les signaux: CMN\_\_Securite\_Signaux

Positionne l'environnement: CMN\_\_Securite\_Environ

Controler les privileges: CMN\_\_Securite\_Init  
-----

LISTE DES PROCEDURES  
-----

E CMN\_\_Securite\_Init

I CMN\_\_Securite\_NoCore

I CMN\_\_Securite\_Signaux

I CMN\_\_Securite\_Environ

### 9.12.2.Conception Détaillée

-----  
Procedure : CMN\_\_Securite\_Init

Description : Initialisation des concepts sécurité en  
appelant les différents modules nécessaires  
CMN\_\_Securite\_NoCore, CMN\_\_Securite\_Signaux,

CMN\_\_Securite\_Environ

Parametres :

(ENTREE) pointeur de fonction arret propre ou NULL

(ENTREE) pointeur de fonction arret rapide ou NULL

Valeur de retour :

Aucune

-----  
Inhibition des cores

Positionnement des signaux

Positionnement des variables d'environnement  
-----

Procédure : CMN\_\_Securite\_NoCore

Description : Inhibe la creation de fichier core en mode non DEBUG

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Positionner la limite de taille du fichier core a 0  
-----

Procédure : CMN\_\_Securite\_Signaux

Description : Inhibe tous les signaux et traite l'appel d'une  
fonction pour SIGTERM et SIGQUIT,SIGILL,SIGSEGV

Parametres :

(ENTREE) pointeur de fonction arret propre

(ENTREE) pointeur de fonction arret rapide

Valeur de retour :

Aucune

-----  
Initialiser la structure signal

Ignorer les signaux normaux

SIGINT,SIGHUP,SIGABRT,SIGCHLD

SI une fonction est donnée pour

l'arrêt normal, la positionner

sur le signal SIGTERM

positionner la procédure sur les signaux

SINON ignorer le signal

FIN SI

SI une fonction est donnée pour

l'arrêt rapide, la positionner

sur les signaux d'urgence

positionner la procédure sur les signaux

SINON ignorer les signaux

FIN SI  
-----

Procédure : CMN\_\_Securite\_Environ

Description : Positionne les variables d'environnement  
Positionnement des signaux  
Inhibition des core

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Démarrage dans le répertoire racine du logiciel  
Positionnement du PATH  
Positionnement IFS SHELL  
Positionnement de umask

## 9.13.CLASSE CMN\_\_STAGETM

### 9.13.1.Spécifications

-----  
Composant : CMN  
CLASSE : CMN\_\_StageTM

Description :  
Cette librairie gère les informations liées aux produits  
genere sur l'exécution d'une commande en provenance du PGF

-----  
LISTE DES METHODES

-----  
E SetStageId  
E GetStageIdStr

### 9.13.2.Conception Détaillée

-----  
Methode : SetStageId

Description : Cette méthode met à jour l'attribut stageId en  
fonction de la chaîne passée en paramètre

Parametres :  
(ENTREE) idStr : stageId

Valeur de retour :  
CMN\_\_STATUT\_OK si la mise à jour de l'attribut a été  
correcte  
CMN\_\_STATUT\_NOK sinon

```
TANT QUE stageId non trouvé
    Compare type enum to id given as parameter
FIN TANT QUE
Compute status execution according to finding result
```

-----

Methode : GetStageIdStr

Description : Renvoie l'attribut stageId sous format ASCII

Parametres :

Aucun

Valeur de retour :

Stage id name sous format ASCII

-----

## 9.14.CLASSE CMN\_\_TRACE

### 9.14.1.Spécifications

-----

Composant : CMN

MODULE : CMN\_\_Trace

Commentaire :

Ce module gère les fonctionnalités de trace dans un fichier  
Les procédures disponibles ici sont utilisées par les macros  
de traces TRINIT,TR0,TR1,TR2,TR3,TR4,TRFIN  
Ce module gère un fichier de trace qui peut être circulaire et  
qui comporte la datation de la trace, la trace elle-même et des  
informations de localisation de l'appel dans les sources.  
Les procédures disponibles sont:  
    CMN\_\_Trace\_Init  
    CMN\_\_Trace\_Tracer  
    CMN\_\_Trace\_Fin  
    CMN\_\_Trace\_Print (interne)

-----

LISTE DES PROCEDURES

-----

```
E CMN__Trace_Init
E CMN__Trace_Fin
E CMN__Trace_Tracer
E CMN__Trace_malloc
E CMN__Trace_realloc
E CMN__Trace_free
I CMN__Trace_Print
```

## 9.14.2.Conception Détaillée

-----  
Procédure : CMN\_\_Trace\_Init

Description : Initialise les paramètres de gestion de traces  
avec les paramètres de la ligne de lancement  
du processus.  
Si argc=0 les traces seront inactives.  
Exploite par getopt les paramètres argv et traduit  
-n x par niveau de trace = x (défaut=0)  
-t fichier\_trace fichier de trace (défaut=  
Trace<pid\_process>)  
-c traces non circulaires (défaut=oui)  
-l ligne du fichier circulaire (défaut=500)  
-s tracé du source et de la ligne (défaut=NON)

Paramètres :

(ENTREE) argc : nombre de paramètres  
(ENTREE) argv : tableau des paramètres

Valeur de retour :

Aucune

-----  
Positionner le nom du fichier de trace par défaut Trace<pid>  
Exploiter les paramètres de lancement  
SELON l'argument passé en paramètre  
CAS 'n': mettre le niveau de trace  
CAS 't': mettre le nom du fichier de trace  
CAS 'c': positionner le flag fichier non circulaire  
CAS 'l': mettre le nombre de ligne si on a un fichier  
circulaire  
CAS 's': positionner le flag de trace des informations  
de source  
FIN SELON  
SI le niveau de trace est différent de 0  
ouvrir le fichier de trace  
tracer dans le fichier de trace les options  
FIN SI

-----  
Procédure : CMN\_\_Trace\_Fin

Description : Termine la gestion des traces en fermant le fichier  
de trace

Paramètres :

Aucun

Valeur de retour :

Aucune

-----  
Fermer le fichier de trace si il a été ouvert  
-----

Procédure : CMN\_\_Trace\_Tracer

Description : Trace dans le fichier de trace les informations

passées. Les paramètres sont :

- le premier est le nom du fichier source de la trace,
- le deuxième est la ligne dans le fichier source,
- le troisième est la chaîne à tracer ou le format de sortie et dans ce cas on a des paramètres pour le format à la suite
- le dernier paramètre est 0 pour fermer les paramètres variable.

La trace est faite dans le fichier avec le format suivant:

YY/MM/JJ hh:mm:ss trace applicative  
fichier:nom\_du\_fichier ligne:numero\_de\_ligne  
(si l'option source est demandée)

Parametres :

- (ENTREE) fichier\_source : fichier C ou C++ appelant la trace
- (ENTREE) numéro\_ligne : ligne dans le fichier C ou C++ appelant la trace
- (ENTREE) ligne : ligne à tracer
- (ENTREE) n\*paramètres : paramètres pour le format (printf like)

Valeur de retour :

Aucune

-----  
Récupérer le pid du process  
Initialisation de la gestion des paramètres variables  
SI le niveau de trace est positif et SI un fichier de trace est ouvert  
    récupérer la date courante  
    récupérer la date courante en milisecondes  
    convertir en date locale  
    convertir la date en chaîne de caractère  
    SI l'information source et ligne est demandée, la mettre dans le format avec la date  
    SINON ne mettre que la date  
    FIN SI  
    Appliquer le format sur les paramètres passés  
    Imprimer dans le fichier de traces  
FIN SI  
Fin de gestion des paramètres variables  
-----

Procédure : CMN\_\_Trace\_malloc

Description : Permet de tracer et de faire un malloc

Parametres :

- (ENTREE) size : taille du malloc
- (ENTREE) source : source appelant
- (ENTREE) ligne : ligne dans le source

Valeur de retour :

Retour du malloc

-----  
Procédure : CMN\_\_Trace\_realloc

Description : Permet de tracer et de faire un realloc

Parametres :

(ENTREE) ptr : adresse existante  
(ENTREE) size : taille du malloc total  
(ENTREE) source : source appellant  
(ENTREE) ligne : ligne dans le source

Valeur de retour :

Retour du realloc

-----  
Procédure : CMN\_\_Trace\_free

Description : Permet de tracer et de faire un free

Parametres :

(ENTREE) adresse a liberer  
(ENTREE) source appellant  
(ENTREE) ligne dans le source

Valeur de retour :

Aucune

-----  
Procédure : CMN\_\_Trace\_Print

Description : Gère l'interface avec le fichier de trace  
pour la gestion du fichier circulaire  
Si la gestion n'est pas circulaire, sauver la ligne  
et c'est tout  
Si s'est du circulaire gérer la position de la ligne  
du fichier pour faire un repositionnement en début  
de fichier quand on arrive à la ligne  
CMN\_\_Trace\_TailleCirculaire

Parametres :

(ENTREE) buffer : buffer à sauver dans le fichier

Valeur de retour :

Aucune

-----  
SI on fonctionne en circulaire voir si on arrive pas à la borne  
pour faire un retour au debut de fichier

FIN SI

Formater la chaine à tracer a un maximun de 255 caractères  
et écrire dans le fichier

Flusher le fichier

## 9.15.CLASSE CMN\_\_VERSION

### 9.15.1.Spécifications

-----  
Composant : CMN

MODULE : CMN\_\_Version

Commentaire :

Ce module permet de gerer la version du logiciel :

- CMN\_\_Version\_Trace
- CMN\_\_Version\_Valeur

-----  
LISTE DES PROCEDURES

-----  
E CMN\_\_Version\_Trace

E CMN\_\_Version\_Valeur

### 9.15.2.Conception Détaillée

-----  
Procédure : CMN\_\_Version\_Trace

Description : Exploite par getopt les paramètres argv et sur -v  
affiche le numéro de version et sort du process

Parametres :

(ENTREE) argc : nombre de paramètres

(ENTREE) argv : tableau des paramètres

Valeur de retour :

Aucune

-----  
Exploiter les paramètres de lancement

SI option 'v', tracer la version et sortir

-----  
Procédure : CMN\_\_Version\_Valeur

Description : Retourne la valeur de la version du logiciel.

Parametres :

Aucun

Valeur de retour :

Pointeur sur la chaine version



## 9.16.CLASSE ERR\_\_EXCEPTION

### 9.16.1.Spécifications

-----

Composant : CMN

CLASSE : ERR\_\_Exception

Commentaire :

Ce fichier gere la classe ERR\_\_Exception

-----

LISTE DES METHODES

-----

E ERR\_\_Exception

E ~ERR\_\_Exception

### 9.16.2.Conception Détaillée

-----

Methode : ERR\_\_Exception

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----

Methode : ~ERR\_\_Exception

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----

## 9.17.CLASSE JDB\_\_CALLBACK

### 9.17.1.Spécifications

-----

Composant : CMN

CLASSE : JDB\_\_Callback

Commentaire :

Ce module gere la classe JDB\_\_Callback qui fournit les méthodes  
appeles par le module C JDB\_JOURNAL\_DE\_BORD pour le traitement  
d'un event de type LOG et TRACE.

-----  
LISTE DES METHODES  
-----

E JDB\_\_Callback  
E ~JDB\_\_Callback  
I JDB\_\_CallbackEVT  
I JDB\_\_CallbackTRA

## 9.17.2.Conception Détaillée

-----

Methode : JDB\_\_Callback

Description : Constructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Initialisation de la liste de callbacks  
Installation des callbacks  
-----

Methode : ~JDB\_\_Callback

Description : Destructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : JDB\_\_CallbackEVT

Description : Callback appelée pour les messages de type Evenement

Parametres :  
(ENTREE) msgId : Id du message  
(ENTREE) objet : Objet du message  
(ENTREE) nivErreur : niveau de criticité du message  
(ENTREE) message : libellé du criticité du message

Valeur de retour :  
Aucune  
-----

Déclaration d'un message Journal de bord  
Récupération et positionnement de la date courante  
Serialisation du message Journal de bord  
Publication du message

-----  
Methode : JDB\_\_CallbackTRA

Description : Callback appelée pour les messages de type Trace

Parametres :

(ENTREE) msgId : Id du message  
(ENTREE) objet : Objet du message  
(ENTREE) nivErreur : niveau de criticité du message  
(ENTREE) message : libellé du criticité du message

Valeur de retour :

Aucune

-----  
Déclaration d'un message Journal de bord  
Récupération et positionnement de la date courante  
Serialisation du message Journal de bord  
Publication du message

## 9.18.CLASSE JDB\_\_CLIENT

### 9.18.1.Spécifications

-----  
Composant : CMN

CLASSE : JDB\_\_Client

Commentaire :

Ce module gere la classe JDB\_\_Client en s'appuyant  
sur le module C JDB\_JOURNAL\_DE\_BORD

-----  
LISTE DES METHODES

-----  
E JDB\_\_Client  
E ~JDB\_\_Client  
E SetParametre  
E EcrireMessage  
E Liberer  
E InitSet

### 9.18.2.Conception Détaillée

-----  
Methode : JDB\_\_Client

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Initialisation de la liste des parametres  
-----

Methode : ~JDB\_\_Client

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Libération de la liste des paramètres  
-----

Methode : SetParametre

Description : Ajoute un paramètre avec sa valeur dans la liste des paramètres

Parametres :

(ENTREE) nom : Nom du parametre.

(ENTREE) valeur : Valeur du parametre.

Valeur de retour :

Aucune

-----  
Ajoute le paramètre dans la liste des paramètres du message  
Envoi d'une exception si le nombre maximum de paramètres a été atteint  
-----

Methode : EcrireMessage

Description : Envoie le message dont le numéro est passé en paramètre au journal de bord

Parametres :

(ENTREE) section : Section contenant le message

(ENTREE) numero : Numero du message

Valeur de retour :

Aucune

-----  
Ecriture du message avec les parametres positionnes  
Emission d'une exception si erreur d'écriture  
-----

Methode : Liberer

Description : Libère la liste des paramètres du message

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Ajoute le paramètre dans la liste des paramètres du message  
-----

Methode : InitSet

Description : Initialise un set de message en les chargeant une fois à vide

Parametres :

(ENTREE) idSet  
(ENTREE) msgDeb  
(ENTREE) msgFinAucun

Valeur de retour :

Aucune

-----  
Initialise le set  
-----

## 9.19.CLASSE JDB\_\_MESSAGE

### 9.19.1.Spécifications

-----  
Composant : CMN

CLASSE : JDB\_\_Message

Commentaire :

Cette classe encapsule le message (au sens MSGS) qui contient l'event (log ou trace) envoyé par les processus au JDBS à travers le MSGS

-----  
LISTE DES METHODES  
-----

E JDB\_\_Message  
E ~JDB\_\_Message  
E Deserialiser  
E Serialiser  
E Serialiser  
E Synthetiser  
E Analyser

## 9.19.2.Conception Détaillée

-----  
Methode : JDB\_\_Message

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : ~JDB\_\_Message

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : Deserialiser

Description : Construit un objet de la classe JDB\_\_Message à partir du contenu du message passé en paramètre

Parametres :

(ENTREE) message : message contenant un message JDB

Valeur de retour :

Aucune

-----  
SI le type du message est bien evenement Journal de bord  
lecture de la version de la classe  
lecture du champ FONCTION  
lecture de l'attribut date dans la section  
lecture de l'attribut msg Id dans la section  
lecture du champ CRITICITE  
lecture du champ LIBELLE  
FIN SI

-----  
Methode : Serialiser

Description : Serialise un objet JDB\_\_Message en un paquet reseau propre à être envoyé

Parametres :

(SORTIE) MSG\_\_Message : Message reseau

Valeur de retour :

Aucune  
-----

Ecriture de la version de la classe  
Ecriture de l'attribut fonction  
Ecriture de l'attribut date dans la section  
Ecriture de l'id du message  
Ecriture de l'attribut criticite  
Ecriture de l'attribut libelle  
Sauvegarde de la serialisation dans une chaine  
Sauvegarde du bloc dans le message  
Sauvegarde du type dans le message  
Positionnement de l'adresse de publication du message si elle  
n'est pas encore renseignée

-----  
Methode : Serialiser

Description : Construit un message a partir de la description de  
l'objet courant

Parametres :

Aucun

Valeur de retour :

Message contenant la description de l'objet

-----  
Declaration d'un message  
Appel de la serialisation  
Retour du message  
-----

Methode : Synthetiser

Description : Génère une section pour l'objet dans le fichier de  
configuration

Parametres :

(ENTREE) config : fichier de configuration  
(ENTREE) numero : numero de la section

Valeur de retour :

Aucune

-----  
Ecriture de l'attribut fonction dans la section  
Récupération et positionnement de la date courante PDS :  
écriture de la date dans la section  
Ecriture de l'attribut fonction Id dans la section  
Ecriture de l'attribut criticite dans la section  
Ecriture de l'attribut libelle en tronquant à gauche  
lorsque le libellé est trop grand  
-----

Methode : Analyser

Description : Analyse une section du fichier de configuration pour  
construire l'objet message

Parametres :

(ENTREE) config : fichier de configuration  
(ENTREE) numero : section decrivant le message

Valeur de retour :  
Aucune

-----  
Lecture du champ FONCTION dans la section  
Lecture de l'attribut date dans la section  
Lecture du champ ID dans la section  
Lecture du champ CRITICITE dans la section  
Lecture du champ LIBELLE dans la section

## 9.20.CLASSE MP\_\_ARRETPROCESS

### 9.20.1.Spécifications

-----  
Composant : MP  
CLASSE : MP\_\_ArretProcess

Commentaire :  
Cette classe encapsule et gère le message de type arret process  
envoyé par les processus applicatifs au MP lorsque ceux-ci  
s'arrête

-----  
LISTE DES METHODES

-----  
E MP\_\_ArretProcess  
E ~MP\_\_ArretProcessus  
E Serialiser  
E Serialiser  
E Deserialiser

### 9.20.2.Conception Détaillée

-----  
Methode : MP\_\_ArretProcess

Description : Constructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : ~MP\_\_ArretProcessus

Description : Destructeur

Parametres :



Aucun

Valeur de retour :

Aucune

-----  
Methode : Serialiser

Description : Serialisation de l'objet

Parametres :

(SORTIE) message : Message reseau.

Valeur de retour :

Aucune

-----  
Ecriture de la version de la classe  
Ecriture du type de PROCESSUS  
Ecriture du type de l'arret du processus  
Sauvegarde de la serialisation dans une chaine  
Sauvegarde du bloc dans le message  
Sauvegarde du type dans le message  
-----

Methode : Serialiser

Description : Serialisation de l'objet

Parametres :

Aucun

Valeur de retour :

message : Message reseau

-----  
Serialisation de l'objet  
Retourne le message  
-----

Methode : Deserialiser

Description : Deserialisation de l'objet

Parametres :

(ENTREE) message : message reseau contenant un objet  
MP\_\_ArretProcessus serialise.

Valeur de retour :

Aucune

-----  
SI le type du message est bien EVT\_ARRET\_PROCESSUS  
lecture de la version de la classe  
lecture du nom du processus émetteur du message  
lecture de l'état de fin du processus  
FIN SI

## 9.21.CLASSE MP\_\_COMMAND

### 9.21.1.Spécifications

-----  
Composant : MP

CLASSE : MP\_\_Command

Commentaire :

Cette classe encapsule et gère le message de type controle  
de la production (step/stop/suspend/resume/break) envoyé par  
le MP au WOM

-----  
LISTE DES METHODES

-----  
E MP\_\_Command

E ~MP\_\_Command

E Serialiser

E Serialiser

E Deserialiser

### 9.21.2.Conception Détaillée

-----  
Methode : MP\_\_Command

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Initialisation des attributs

-----  
Methode : ~MP\_\_Command

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : Serialiser

Description : Serialisation de l'objet

## Parametres :

(ENTREE) typeMessage : type du message CMD ou CMD\_TM  
(SORTIE) message : Message reseau

## Valeur de retour :

Aucune

-----  
Ecriture de la version de la classe  
Ecriture des informations générales d'une commande :  
- le groupe  
- l'identificateur de la cmde

Ecriture des informations de la commande :  
- le type de commande  
- le status de la commande  
- le nom du work order

Ecriture des informations propres au message échangé entre  
le MP et le WOM : l'Id du processus SD  
Sauvegarde de la serialisation dans une chaine  
Sauvegarde du bloc dans le message  
Sauvegarde du type dans le message  
-----

## Methode : Serialiser

Description : Serialisation de l'objet

## Parametres :

Aucun

## Valeur de retour :

message : Message reseau

-----  
Serialisation de l'objet  
Retourne le message  
-----

## Methode : Deserialiser

Description : Deserialisation de l'objet

## Parametres :

(ENTREE) message : message reseau contenant un objet  
MP\_\_Command serialisé

## Valeur de retour :

Aucune

-----  
SI le type du message est bien EVT\_PRCC\_CTRL ou EVT\_\_CMD\_TM  
lecture de la version de la classe  
lecture des informations générales d'une commande :  
- le groupe  
- l'identificateur de la cmde  
lecture des informations de la commande :  
- le type de commande  
- le status de la commande  
- le nom du work order  
lecture des informations propres au message échangé entre

le MP et le WOM : l'Id du processus SD  
FIN SI

## 9.22.CLASSE MP\_\_DUMP

### 9.22.1.Spécifications

-----  
Composant : MP

CLASSE : MP\_\_Dump

Commentaire :

Cette classe encapsule et gère le message de type fin de traitement d'un dump envoyé par le processus WOM au MP lorsque le traitement d'un dump est terminé

-----  
LISTE DES METHODES

-----  
E MP\_\_Dump

E ~MP\_\_Dump

E Serialiser

E Serialiser

E Deserialiser

### 9.22.2.Conception Détaillée

-----  
Methode : MP\_\_Dump

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : ~MP\_\_Dump

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : Serialiser

Description : Serialisation de l'objet

Parametres :

(SORTIE) message : Message reseau

Valeur de retour :

Aucune

-----  
Ecriture de la version de la classe  
Ecriture de l'etat de fin de traitement du dump courant  
Sauvegarde de la serialisation dans une chaine  
Sauvegarde du bloc dans le message  
Sauvegarde du type dans le message  
-----

Methode : Serialiser

Description : Serialisation de l'objet

Parametres :

Aucun

Valeur de retour :

message : Message reseau

-----  
Serialisation de l'objet  
Retourne le message  
-----

Methode : Deserialiser

Description : Deserialisation de l'objet

Parametres :

(ENTREE) message : message reseau contenant un objet  
MP\_\_ArretProcessus serialisé

Valeur de retour :

Aucune

-----  
SI le type du message est bien EVT\_FIN\_DUMP  
lecture de la version de la classe  
lecture de l'état de fin de traitement du dump  
FIN SI  
-----

## 9.23.CLASSE MP\_\_HKTMSTATUS

### 9.23.1.Spécifications

-----  
Composant : MP

CLASSE : MP\_\_HKTMStatus

Commentaire :

Cette classe gère les HKTM status échangés entre les processus applicatifs et le MP

-----  
LISTE DES METHODES  
-----

E MP\_\_HKTMStatus  
E ~MP\_\_HKTMStatus  
E Serialiser  
E Serialiser  
E Deserialiser  
E GetParamName  
E GetValueString

## 9.23.2.Conception Détaillée

-----

Methode : MP\_\_HKTMStatus

Description : Constructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Initialisation des attributs à des valeurs par défaut  
-----

Methode : ~MP\_\_HKTMStatus

Description : Destructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : Serialiser

Description : Serialisation de l'objet

Parametres :  
(SORTIE) message : Message reseau.

Valeur de retour :  
Aucune

-----  
Ecriture de la version de la classe

Ecriture de l'Id de la HKTM  
Ecriture de la validite  
Ecriture de la disponibilite  
Ecriture du service emetteur des données (SPACECRAFT)  
Ecriture du type de produit en cours de génération  
Ecriture de la valeur chaine de la ressource  
Sauvegarde de la serialisation dans une chaine  
Sauvegarde du bloc dans le message  
Sauvegarde du type dans le message  
Positionnement de l'adresse de publication du message si elle  
n'est pas encore renseignée

-----  
Methode : Serialiser

Description : Serialisation de l'objet

Parametres :

Aucun

Valeur de retour :

message : Message reseau

-----  
Serialisation de l'objet  
Retourne le message  
-----

Methode : Deserialiser

Description : Deserialisation de l'objet

Parametres :

(ENTREE) message : message reseau contenant un objet  
MP\_\_HKTMStatus serialise.

Valeur de retour :

Aucune

-----  
SI le type du message est bien EVT\_HKTM\_STATUS  
lecture de la version de la classe  
lecture de l'Id de la HKTM  
lecture de la validité  
lecture de la disponibilité  
lecture du service émetteur des données (SPACECRAFT)  
lecture du type de produit en cours de generation  
lecture de la valeur chaine de la ressource  
FIN SI  
-----

Methode : GetParamName

Description : Construit et renvoie le nom du HKTM status courant  
géré par l'objet.

Parametres :

(SORTIE) paramName : nom du HKTM status courant  
géré par l'objet et passé à la  
fonction mla\_SendFunctionStatus

Valeur de retour :

Aucune

-----  
Le nom du HKTM Status est formé de la concatenation des champs :

- Facility => constante
- Category => dépend du type de HKTM
- Location => constante
- Type => dépend du type de HKTM
- Service => mis à jour par le WOM a la réception  
du lier fichier de données
- Id => dépend du type de HKTM et du produit courant

-----  
Methode : GetValueString

Description : Construit et renvoie la valeur du HKTM status courant  
géré par l'objet.

Parametres :

(SORTIE) paramString : valeur du HKTM status courant  
géré par l'objet et passée à la  
fonction mla\_SendFunctionStatus

Valeur de retour :

Aucune

-----  
Affecter la valeur stockée dans l'attribut dédié de l'objet

## 9.24.CLASSE MP\_\_SPACECRAFT

### 9.24.1.Spécifications

-----  
Composant : MP

CLASSE : MP\_\_Spacecraft

Commentaire :

Ce module gère la classe MP\_\_Spacecraft

-----  
LISTE DES METHODES

-----  
E MP\_\_Spacecraft  
E ~MP\_\_Spacecraft  
E Serialiser  
E Serialiser  
E Deserialiser  
E SetSpacecraft  
E GetSpacecraftStr



## 9.24.2. Conception Détaillée

-----  
Methode : MP\_\_Spacecraft

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : ~MP\_\_Spacecraft

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : Serialiser

Description : Serialisation de l'objet

Parametres :

(SORTIE) message : Message reseau.

Valeur de retour :

Aucune

-----  
Ecriture de la version de la classe  
Ecriture du type SPACECRAFT  
Sauvegarde de la serialisation dans une chaine  
Sauvegarde du bloc dans le message  
Sauvegarde du type dans le message  
-----

Methode : Serialiser

Description : Serialisation de l'objet

Parametres :

Aucun

Valeur de retour :

message : Message reseau

-----  
Serialisation de l'objet  
Retourne le message  
-----

Methode : Deserialiser

Description : Deserialisation de l'objet

Parametres :

(ENTREE) message : message reseau contenant un objet  
MP\_\_Spacecraft serialise.

Valeur de retour :

Aucune

```
-----
SI le type du message est bien EVT_SPACECRAFT
  lecture de la version de la classe
  lecture du Spacecraft
FIN SI
-----
```

Methode : SetSpacecraft

Description : This method stores the spacecraft according to the  
string given as input parameter

Parametres :

(ENTREE) spacecraft name

Valeur de retour :

CMN\_\_STATUT\_OK si l'attribut a été mis à jour  
CMN\_\_STATUT\_NOK sinon

```
-----
TANT QUE service type not found
  Compare type enum to spacecraft type given as parameter
FIN TANT QUE
Compute status execution according to finding result
-----
```

Methode : GetSpacecraftStr

Description : Provide the current spcacraft in ASCII format

Parametres :

Aucun

Valeur de retour :

Spacecraft in ASCII format

## 9.25.CLASSE MP\_\_STAGETM

### 9.25.1.Spécifications

Composant : MP

CLASSE : MP\_\_StageTM

Commentaire :

Cette classe encapsule et gere les informations echangées entre

le MP et le WOM liées aux fichiers de données produits.

-----  
LISTE DES METHODES  
-----

E Serialiser  
E Serialiser  
E Deserialiser

## 9.25.2.Conception Détaillée

-----

Methode : Serialiser

Description : Serialisation de l'objet

Parametres :  
(SORTIE) message : Message reseau

Valeur de retour :  
Aucune

-----  
Ecriture de la version de la classe  
Ecriture des informations générales d'une commande :  
- le groupe  
- l'identificateur de la cmde  
Ecriture des informations de la commande :  
- l'identificateur de la commande StageTM  
- la category du produit  
- le nom du fichier produit  
Sauvegarde de la serialisation dans une chaine  
Sauvegarde du bloc dans le message  
Sauvegarde du type dans le message  
-----

Methode : Serialiser

Description : Serialisation de l'objet

Parametres :  
Aucun

Valeur de retour :  
message : Message reseau

-----  
Serialisation de l'objet  
Retourne le message  
-----

Methode : Deserialiser

Description : Deserialisation de l'objet

Parametres :  
(ENTREE) message : message reseau contenant un objet

MP\_\_Command serialise

Valeur de retour :  
Aucune

-----  
SI le type du message est bien EVT\_STAGE\_TM  
  lecture de la version de la classe  
  lecture des informations générales d'une commande :  
    - le groupe  
    - l'identificateur de la cmde  
  lecture des informations de la commande :  
    - l'identificateur de la commande StageTM  
    - la category du produit  
    - le nom du fichier produit  
FIN SI

## 9.26.CLASSE MP\_\_UNPROCESS

### 9.26.1.Spécifications

-----  
Composant : MP  
CLASSE : MP\_\_UnProcess

Commentaire :  
  Ce fichier gere la classe MP\_\_UnProcess

-----  
LISTE DES METHODES  
-----

E Deserialiser  
E Serialiser  
E Serialiser

### 9.26.2.Conception Détaillée

-----  
Methode : Deserialiser

Description : Construit un objet de la classe MP\_\_UnProcess à  
partir du contenu du message passé en paramètre

Parametres :  
  (ENTREE) message : message contenant un process

Valeur de retour :  
Aucune

-----  
SI le type du message est bien evenement PROCESS  
  lecture de la version de la classe

lecture de l'action demandée  
lecture du process id  
lecture du nom du process

FIN SI

-----  
Methode : Serialiser

Description : Construit un message a partir de la description de  
l'objet courant

Parametres :

(SORTIE) MSG\_\_Message : Message reseau

Valeur de retour :

Aucune

-----  
Ecriture de la version de la classe  
Ecriture de la valeur de l'action demandee  
Ecriture de la valeur du process id  
Ecriture du nom du process  
Sauvegarde de la serialisation dans une chaine  
Sauvegarde du bloc dans le message  
Sauvegarde du type dans le message  
Positionnement de l'adresse de publication du message s'il  
n'est pas encore renseigne  
-----

Methode : Serialiser

Description : Construit un message a partir de la description de  
l'objet courant

Parametres :

Aucun

Valeur de retour :

Message contenant la description de l'objet

-----  
Déclaration d'un message  
Appel de la serialisation  
Retour du message

## 9.27.CLASSE SD\_\_INIT

### 9.27.1.Spécifications

-----  
Composant : CMN

CLASSE : SD\_\_Init

Commentaire :

Ce fichier contient la classe SD\_\_Init, qui permet de gérer le  
message de fin d'initialisation du composant SD

-----  
LISTE DES METHODES  
-----

E SD\_\_Init  
E ~SD\_\_Init  
E Deserialiser  
E Serialiser  
E Serialiser

## 9.27.2.Conception Détaillée

  
-----

Methode : SD\_\_Init

Description : Constructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : ~SD\_\_Init

Description : Destructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : Deserialiser

Description : Construit un objet de la classe SD\_\_Init à partir  
du contenu du message passe en parametre

Parametres :  
(ENTREE) message : Message contenant l'objet

Valeur de retour :  
Aucune

-----  
SI le type du message est bien événement INIT\_SD  
lecture de la version de la classe  
lecture de l'etat de l'initialisation  
FIN SI  
-----

Methode : Serialiser

Description : Construit un message à partir de la description de

l'objet courant

Parametres :

(SORTIE) MSG\_\_Message : Message réseau

Valeur de retour :

Aucune

-----  
Ecriture de la version de la classe  
Ecriture du type de traitement  
Sauvegarde de la serialisation dans une chaine  
Sauvegarde du bloc dans le message  
Sauvegarde du type dans le message  
Positionnement de l'adresse de publication du message si  
elle n'est pas encore renseignée  
-----

Methode : Serialiser

Description : Construit un message à partir de la description de  
l'objet courant

Parametres :

Aucun

Valeur de retour :

Message contenant la description de l'objet

-----  
Déclaration d'un message  
Appel de la serialisation  
Retour du message  
-----

## 9.28.CLASSE TES\_\_EVENEMENT

### 9.28.1.Spécifications

-----  
Composant : TES

CLASSE : TES\_\_Evenement

Commentaire :

Ce fichier gère la classe TES\_\_Evenement

-----  
LISTE DES METHODES  
-----

E TES\_\_Evenement  
E ~TES\_\_Evenement  
E Serialiser  
E Serialiser  
E Deserialiser

## 9.28.2.Conception Détaillée

-----  
Methode : TES\_\_Evenement

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : ~TES\_\_Evenement

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : Serialiser

Description : Serialise un objet TES\_\_Evnement en un paquet reseau  
propre a etre envoye

Parametres :

(SORTIE) MSG\_\_Message : Message reseau

Valeur de retour :

Aucune

-----  
Ecriture de la version de la classe

Ecriture de la valeur de l'action demandee

Ecriture de la periode du timer

Ecriture de la date de debut

Ecriture du nom logique

Ecriture du type logique

Ecriture de l'adresse

Ecriture des donnees utilisateur

Sauvegarde de la serialisation dans une chaine

Sauvegarde du bloc dans le message

Sauvegarde du type dans le message

Positionnement de l'adresse de publication du message si elle  
n'est pas encore renseignée

-----  
Methode : Serialiser

Description : Serialise un objet TES\_\_Evnement en un paquet reseau  
propre a etre envoye

Parametres :

Aucun



Valeur de retour :

MSG\_\_Message : Message reseau

-----  
Déclaration d'un message  
Appel de la serialisation  
Retour du message  
-----

Methode : Deserialiser

Description : Construit un objet TES\_\_Evenement a partir d'un  
paquet reseau

Parametres :

(ENTREE) MSG\_\_Message : Message reseau

Valeur de retour :

Aucune

-----  
SI le type du message est bien evenement TIMER alors  
lecture de la version de la classe  
lecture de l'action demandée  
lecture de la période du timer  
lecture de la date de debut  
lecture du nom logique  
lecture du type logique  
lecture de l'adresse  
lecture des données utilisateur  
FIN SI

## 9.29.CLASSE CMN\_DAT\_EVENTEMENT

### 9.29.1.Spécifications

-----  
Composant : CMN

Sous-composant : CMN\_DAT

CLASSE : CMN\_DAT\_Evenement

Commentaire :

Ce module gère la classe CMN\_DAT\_Evenement, qui permet de  
stocker l'ensemble des événements

-----  
LISTE DES METHODES  
-----

E CMN\_DAT\_Evenement  
E CMN\_DAT\_Evenement  
E ~CMN\_DAT\_Evenement  
E operator =  
E Synthetiser  
E Analyser

## 9.29.2.Conception Détaillée

-----  
Methode : CMN\_DAT\_Evenement

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : CMN\_DAT\_Evenement

Description : Constructeur copie

Parametres :

(ENTREE) unEvenement

Valeur de retour :

Aucune

-----  
Methode : ~CMN\_DAT\_Evenement

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : operator =

Description : Operateur =

Parametres :

(ENTREE) unEvenement

Valeur de retour :

CMN\_DAT\_Evenement : Evénement mis à jour

-----  
Methode : Synthetiser

Description : Génère une section pour l'événement dans le fichier de configuration

Parametres :

(ENTREE) Config : Fichier de configuration

Valeur de retour :  
Aucune

-----  
Génération de la section  
On sauvegarde :  
- la date de l'événement  
- le type d'événement  
- l'action ayant produit l'événement  
- le libellé de l'événement  
-----

Methode : Analyser

Description : Analyse une section du fichier de configuration pour  
remplir l'événement

Parametres :  
(ENTREE) Config : Fichier de configuration  
(ENTREE) sNomEvenement: Section du fichier de  
configuration décrivant  
l'événement

Valeur de retour :  
Aucune

-----  
Analyse de la section  
On recupère :  
- la date de l'événement  
- le type d'événement  
- l'action ayant produit l'événement  
- le libellé de l'événement  
-----

## 9.30.CLASSE CMN\_DAT\_LISTEOBJETS

### 9.30.1.Spécifications

-----  
Composant : CMN  
Sous-composant : CMN\_DAT  
CLASSE : CMN\_DAT\_ListeObjets

Commentaire :  
Ce module gère la classe CMN\_DAT\_ListeObjets

-----  
LISTE DES METHODES  
-----

E CMN\_DAT\_ListeObjets  
E CMN\_DAT\_ListeObjets  
E ~CMN\_DAT\_ListeObjets  
E Contenu  
E operator =  
E CompterElements  
E LireIdentifiantElement

E LireElement  
E RechercherElement  
E AjouterElement  
E SupprimerElement  
E ChargerListe  
E SauverListe

### 9.30.2.Conception Détaillée

-----  
Methode : CMN\_DAT\_ListeObjets

Description : Constructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : CMN\_DAT\_ListeObjets

Description : Constructeur

Parametres :  
(ENTREE) NomFichier

Valeur de retour :  
Aucune

-----  
Methode : ~CMN\_DAT\_ListeObjets

Description : Destructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : Contenu

Description : Retourne le contenu de l'objet sous forme de chaine

Parametres :  
Aucun

Valeur de retour :  
Contenu sous forme de chaine de caractere

Methode : operator =

Description : operateur =

Parametres :

(ENTREE) ListeObjets : Liste à affecter

Valeur de retour :

Aucune

-----  
Methode : CompterElements

Description : Compte le nombre d'éléments dans la liste

Parametres :

Aucun

Valeur de retour :

Nombre d'éléments dans la liste (-1 si erreur)

-----  
Methode : LireIdentifiantElement

Description : Récupère l'identifiant d'un objet à partir de sa position

Parametres :

(ENTREE) nPosition: Position de l'objet dans la liste  
(SORTIE) Identifiant: Identifiant de l'objet se trouvant à cette position dans la liste

Valeur de retour :

CMN\_\_STATUT\_OK si la fonction a réussi  
CMN\_\_STATUT\_NOK si la fonction a échoué

-----  
Methode : LireElement

Description : Lit un élément de la liste à partir de sa position

Parametres :

(ENTREE) nPosition: Position de l'objet à lire dans la liste  
(SORTIE) Objet: Objet se trouvant à cette position dans la liste

Valeur de retour :

CMN\_\_STATUT\_OK si la fonction a réussi  
CMN\_\_STATUT\_NOK si la fonction a échoué

-----  
On initialise l'objet pour que les informations susceptibles d'être contenues dans l'Objet en entrée ne perturbent pas le résultat

-----  
Methode : RechercherElement

Description : Recherche un élément de la liste à partir de l'identifiant

Parametres :

(ENTREE) sIdentifiant : Identifiant de l'objet à rechercher

Valeur de retour :

Position de l'élément dans la liste (-1 si erreur)

-----  
Methode : AjouterElement

Description : Ajoute un élément dans la liste

Parametres :

(ENTREE) Objet : Objet à ajouter dans la liste

Valeur de retour :

Position de l'objet ajouté (-1 si erreur)

-----  
Methode : SupprimerElement

Description : Supprime un élément de la liste

Parametres :

(ENTREE) nPosition : Position de l'objet à supprimer dans la liste

Valeur de retour :

CMN\_\_STATUT\_OK si la fonction a réussi  
CMN\_\_STATUT\_NOK si la fonction a échoué

-----  
Methode : ChargerListe

Description : Charge la liste à partir d'un fichier

Parametres :

(ENTREE) NomFichier : Nom du fichier contenant la description des objets

Valeur de retour :

CMN\_\_STATUT\_OK si la fonction a réussi  
CMN\_\_STATUT\_NOK si la fonction a échoué

-----  
Methode : SauverListe

Description : Sauve la liste dans un fichier

Parametres :

(ENTREE) NomFichier : Nom du fichier qui contiendra la description des objets

Valeur de retour :

CMN\_\_STATUT\_OK si la fonction a réussi  
CMN\_\_STATUT\_NOK si la fonction a échoué

## 9.31.CLASSE CMN\_DAT\_MODELEACTION

### 9.31.1.Spécifications

Composant : CMN

Sous-composant : CMN\_DAT

CLASSE : CMN\_DAT\_ModeleAction

Commentaire :

Ce module gère la classe CMN\_DAT\_ModeleAction, qui permet de stocker l'ensemble des actions

LISTE DES METHODES

E CMN\_DAT\_ModeleAction()  
E CMN\_DAT\_ModeleAction()  
E ~CMN\_DAT\_ModeleAction()  
E operator =  
E Analyser

### 9.31.2.Conception Détaillée

Methode : CMN\_DAT\_ModeleAction()

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

Methode : CMN\_DAT\_ModeleAction()

Description : Constructeur

Parametres :

(ENTREE) unModeleAction

Valeur de retour :

Aucune

-----  
 Methode : ~CMN\_DAT\_ModeleAction()

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
 Methode : operator =

Description : Operateur =

Parametres :

(ENTREE) unModeleAction : Objet source

Valeur de retour :

CMN\_DAT\_ModeleAction : Objet destination

-----  
 Methode : Analyser

Description : Analyse une section du fichier de configuration pour  
remplir le ModeleAction

Parametres :

(ENTREE) Config : Fichier de configuration

(ENTREE) sIdentifiantElement: Section du fichier de  
configuration décrivant  
le modèle des actions

Valeur de retour :

Aucune

-----  
 On lit le temps de simulation

## 9.32.CLASSE CMN\_DAT\_MODELETACHE

### 9.32.1.Spécifications

-----  
 Composant : CMN

Sous-composant : CMN\_DAT

CLASSE : CMN\_DAT\_ModeleTache

Commentaire :

Ce module gère la classe CMN\_DAT\_ModeleTache, qui permet de  
stocker l'ensemble des taches

-----  
 LISTE DES METHODES



```
-----  
E CMN_DAT_ModeleTache()  
E CMN_DAT_ModeleTache()  
E ~CMN_DAT_ModeleTache()  
E operator =  
E Analyser  
E Action
```

### 9.32.2.Conception Détaillée

```
-----  
Methode : CMN_DAT_ModeleTache()
```

Description : Constructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

```
-----  
Methode : CMN_DAT_ModeleTache()
```

Description : Constructeur

Parametres :  
(ENTREE) unModeleTache

Valeur de retour :  
Aucune

```
-----  
Methode : ~CMN_DAT_ModeleTache()
```

Description : Destructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

```
-----  
Methode : operator =
```

Description : Operateur =

Parametres :  
(ENTREE) unModeleTache : Objet source

Valeur de retour :  
CMN\_DAT\_ModeleTache : Objet destination

-----  
Methode : Analyser

Description : Analyse une section du fichier de configuration pour  
remplir le ModeleTache

Parametres :

(ENTREE) Config : Fichier de configuration  
(ENTREE) sIdentifiantElement: Section du fichier de  
configuration décrivant  
le modèle de taches

Valeur de retour :

Aucune

-----  
On lit l'occurrence  
On lit le temps de simulation  
On lit le nombre d'actions  
Sauvegarde des actions  
-----

Methode : Action

Description : Retourne une action

Parametres :

(ENTREE) NoAction (0 à m\_NombreActions-1): Numéro de  
l'action à  
retourner

Valeur de retour :

Aucune

## 9.33.CLASSE CMN\_DAT\_MODELETRAITEMENT

### 9.33.1.Spécifications

-----  
Composant : CMN

Sous-composant : CMN\_DAT

CLASSE : CMN\_DAT\_ModeleTraitement

Commentaire :

Ce module gère la classe CMN\_DAT\_ModeleTraitement, qui permet  
de stocker l'ensemble des traitements

-----  
LISTE DES METHODES  
-----

E CMN\_DAT\_ModeleTraitement()  
E CMN\_DAT\_ModeleTraitement  
E ~CMN\_DAT\_ModeleTraitement()  
E operator =

E Analyser  
E Tache  
E TypeTache

### 9.33.2.Conception Détaillée

-----  
Methode : CMN\_DAT\_ModeleTraitement()

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : CMN\_DAT\_ModeleTraitement

Description : Constructeur

Parametres :

(ENTREE) unModeleTraitement

Valeur de retour :

Aucune

-----  
Methode : ~CMN\_DAT\_ModeleTraitement()

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : operator =

Description : Operateur =

Parametres :

(ENTREE) unModeleTraitement : Objet source

Valeur de retour :

CMN\_DAT\_ModeleTraitement : Objet destination

-----  
Methode : Analyser

Description : Analyse une section du fichier de configuration pour

remplir le modèle de traitement

Parametres :

(ENTREE) Config : Fichier de configuration  
(ENTREE) sIdentifiantElement: Section du fichier de  
configuration décrivant  
le modeleTraitement

Valeur de retour :

Aucune

-----  
On lit le nombre de traitements  
Sauvegarde des taches  
-----

Methode : Tache

Description : Retourne une tache

Parametres :

(ENTREE) NoTache (0 à m\_NombreTaches-1): Numéro de  
la tache à  
retourner

Valeur de retour :

Aucune

-----  
Methode : TypeTache

Description : Retourne le type d'une tache

Parametres :

(ENTREE) NoTache (0 à m\_NombreTaches-1): Numéro de  
la tache à  
retourner

Valeur de retour :

Aucune  
-----

## 9.34.CLASSE CMN\_DAT\_OBJET

### 9.34.1.Spécifications

-----  
Composant : CMN

Sous-composant : CMN\_DAT

CLASSE : CMN\_DAT\_Objct

Commentaire :

Ce module gère la classe CMN\_DAT\_Objct  
-----

## LISTE DES METHODES

-----  
E CMN\_DAT\_Objeto()  
E ~CMN\_DAT\_Objeto()  
E operator =  
E Synthetiser  
E Analyser  
E ChargerFichier  
E SauverFichier  
I BouleenVersChaine  
I ChaineVersBouleen  
I EntierVersChaine  
I ChaineVersEntier  
I ReelVersChaine  
I ChaineVersReel

## 9.34.2.Conception Détaillée

-----  
Methode : CMN\_DAT\_Objeto()

Description : Constructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : ~CMN\_DAT\_Objeto()

Description : Destructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : operator =

Description : Operateur =

Parametres :  
(ENTREE) unObjeto : Objeto origine

Valeur de retour :  
CMN\_DAT\_Objeto : Objeto destination

-----  
Methode : Synthetiser

Description : Génère une section pour l'objet dans le fichier de configuration

Parametres :

(ENTREE) Config : Fichier de configuration

Valeur de retour :

Aucune

-----  
On n'écrit rien car l'identifiant est le nom de la section  
qui est créé lorsque l'on enregistre un de ses champs  
-----

Methode : Analyser

Description : Analyse une section du fichier de configuration pour remplir l'objet

Parametres :

(ENTREE) Config : Fichier de configuration  
(ENTREE) sIdentifiantElement: Section du fichier  
de configuration  
décrivant l'objet

Valeur de retour :

Aucune

-----  
Pour l'objet de base, l'identifiant est le nom de la section  
Lire la valeur  
-----

Methode : ChargerFichier

Description : Charge la liste à partir d'un fichier

Parametres :

(ENTREE) NomFichier : Nom du fichier contenant la  
description des objets

Valeur de retour :

CMN\_\_STATUT\_OK si le chargement a été réalisé  
CMN\_\_STATUT\_NOK sinon

-----  
Methode : SauverFichier

Description : Sauve l'objet dans un fichier

Parametres :

(ENTREE) NomFichier du fichier qui contiendra l'objet

Valeur de retour :

CMN\_\_STATUT\_OK si la sauvegarde a été réalisée  
CMN\_\_STATUT\_NOK sinon

-----  
Methode : BouleenVersChaine

Description : Convertit une valeur booléenne en chaîne de caractère  
(utilisé pour la synthèse d'un objet)

Parametres :

(ENTREE) Valeur booléenne : 0 : OUI, 1: NON

Valeur de retour :

Chaîne contenant la valeur

-----  
Methode : ChaineVersBouleen

Description : Convertit une chaîne de caractère en valeur booléenne  
(utilisé pour l'analyse d'un objet)

Parametres :

(ENTREE) Valeur booléenne : 0 : OUI 1: NON

Valeur de retour :

Booleen contenant la valeur

-----  
Methode : EntierVersChaine

Description : Convertit une valeur entière en chaîne de caractère  
(utilisé pour la synthèse d'un objet)

Parametres :

(ENTREE) Valeur : Valeur à convertir

Valeur de retour :

Chaîne contenant la valeur

-----  
Methode : ChaineVersEntier

Description : Convertit une chaîne de caractère en valeur entière  
(utilisé pour l'analyse d'un objet)

Parametres :

(ENTREE) s: Valeur à convertir

Valeur de retour :

Nombre contenant la valeur

-----  
Methode : ReelVersChaine

Description : Convertit une valeur réelle en chaîne de caractère  
(utilisé pour la synthèse d'un objet)

Parametres :

(ENTREE) Valeur : Valeur à convertir

Valeur de retour :

Chaîne contenant la valeur

-----

Methode : ChaineVersReel

Description : Convertit une chaine de caractère en valeur réelle  
(utilisé pour l'analyse d'un objet)

Parametres :  
(ENTREE) s: Valeur à convertir

Valeur de retour :  
Nombre contenant la valeur

## 9.35.CLASSE CMN\_DAT\_WORKORDER

### 9.35.1.Spécifications

Composant : CMN  
Sous-composant : CMN\_DAT  
CLASSE : CMN\_DAT\_WorkOrder

Commentaire :  
Ce module gère la classe CMN\_DAT\_WorkOrder, qui stocke les  
informations liées à un Work Order

#### LISTE DES METHODES

E CMN\_DAT\_WorkOrder  
E ~CMN\_DAT\_WorkOrder  
E ProcessingType  
E Afficher  
I TransformeDate

### 9.35.2.Conception Détaillée

Methode : CMN\_DAT\_WorkOrder

Description : Constructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

Methode : ~CMN\_DAT\_WorkOrder



Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : ProcessingType

Description : Permet de mettre à jour l'attribut ProcessingType

Parametres :

(ENTREE) type : Chaîne contenant le type

Valeur de retour :

CMN\_\_STATUT\_OK : Type valide

CMN\_\_STATUT\_NOK : Type invalide

-----  
Convertir la chaîne en majuscules

TANT QUE stage id non trouvé

Comparer le type et l'identifiant passé en paramètre

FIN TANT QUE

-----  
Methode : Afficher

Description : Affiche le contenu de l'objet

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : TransformeDate

Description : Permet de transformer une date UTC sous forme d'une chaîne de caractères en objet Date

Parametres :

(ENTREE) dateIn : Chaîne contenant la date

(SORTIE) dateOut : Objet date

Valeur de retour :

CMN\_\_STATUT\_OK : Format de la date valide

CMN\_\_STATUT\_NOK : Format de la date invalide

-----  
Recherche des fractions de seconde

SI le format ne spécifie pas les fractions de secondes

SINON le format spécifie les fractions de secondes

FIN SI

## 9.36.CLASSE CMN\_OBS\_OBSERVATEUR

### 9.36.1.Spécifications

-----  
Composant : CMN  
Sous-composant : CMN\_OBS  
CLASSE : CMN\_OBS\_Observateur

Commentaire :  
Ce module gère la classe CMN\_OBS\_Observateur

#### ----- LISTE DES METHODES

-----  
E CMN\_OBS\_Observateur  
E ~CMN\_OBS\_Observateur  
E MettreAJour

### 9.36.2.Conception Détaillée

-----  
Methode : CMN\_OBS\_Observateur

Description : Constructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : ~CMN\_OBS\_Observateur

Description : Destructeur

Parametres :  
Aucun

Valeur de retour :  
Aucune

-----  
Methode : MettreAJour

Description : Demande de mise à jour de l'observateur par le sujet

Parametres :  
(ENTREE) CMN\_OBS\_Sujet \* unSujet : Sujet origine  
de l'appel

Valeur de retour :

Aucune

-----

## 9.37.CLASSE CMN\_OBS\_SUJET

### 9.37.1.Spécifications

-----

Composant : CMN

Sous-composant : CMN\_OBS

CLASSE : CMN\_OBS\_Sujet

Commentaire :

Ce module gère la classe CMN\_OBS\_Sujet

-----

LISTE DES METHODES

-----

E CMN\_OBS\_Sujet

E ~CMN\_OBS\_Sujet

E Attacher

E Detacher

E Notifie

### 9.37.2.Conception Détaillée

-----

Methode : CMN\_OBS\_Sujet

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----

Methode : ~CMN\_OBS\_Sujet

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----

Methode : Attacher

Description : Associe un observateur au sujet

Parametres :

(ENTREE) Observateur : Observateur à attacher

Valeur de retour :

Aucune

-----  
Methode : Detacher

Description : Detache un observateur du sujet

Parametres :

(ENTREE) Observateur : Observateur à détacher

Valeur de retour :

Aucune

-----  
Methode : Notifie

Description : Notifie les observateurs du sujet

Parametres :

Aucun

Valeur de retour :

Aucune

## 9.38.CLASSE CMN\_PRS\_DUMPFORMATTARGET

### 9.38.1.Spécifications

-----  
Composant : CMN

Sous-composant : CMN\_PRS

CLASSE : CMN\_PRS\_DumpFormatTarget

Commentaire :

Ce fichier gère la classe CMN\_PRS\_DumpFormatTarget

-----  
LISTE DES METHODES

-----  
E CMN\_PRS\_DumpFormatTarget  
E CMN\_PRS\_DumpFormatTarget  
E ~CMN\_PRS\_DumpFormatTarget  
E writeChars

## 9.38.2.Conception Détaillée

-----  
Methode : CMN\_PRS\_DumpFormatTarget

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : CMN\_PRS\_DumpFormatTarget

Description : Constructeur

Parametres :

(ENTREE) filename : Nom du fichier de sortie

Valeur de retour :

Aucune

-----  
Methode : ~CMN\_PRS\_DumpFormatTarget

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Methode : writeChars

Description : Ecrit dans le fichier de sortie

Parametres :

(ENTREE) toWrite

(ENTREE) count

(ENTREE) formatter

Valeur de retour :

Aucune

## 9.39.CLASSE CMN\_PRS\_ERRORREPORTER

### 9.39.1.Spécifications

-----  
Composant : CMN

CLASSE : CMN\_PRS\_ErrorReporter

Commentaire :

Ce fichier gère la classe CMN\_PRS\_ErrorReporter

-----  
LISTE DES METHODES  
-----

E warning  
E error  
E fatalError  
E resetErrors

## 9.39.2.Conception Détaillée

-----

Methode : warning

Description : Gestion des warnings

Parametres :

(ENTREE) toCatch

Valeur de retour :

Aucune

-----  
Methode : error

Description : Affiche une erreur

Parametres :

(ENTREE) toCatch

Valeur de retour :

Aucune

-----  
Methode : fatalError

Description : Affiche une erreur fatale

Parametres :

(ENTREE) toCatch

Valeur de retour :

Aucune

-----  
Methode : resetErrors

Description : Reinitialise les erreurs

Parametres :

Aucun

Valeur de retour :

Aucune

## 9.40.CLASSE CMN\_PRS\_PARSER

### 9.40.1.Spécifications

Composant : CMN

CLASSE : CMN\_PRS\_Parser

Commentaire :

Ce fichier gère la classe CMN\_PRS\_Parser

LISTE DES METHODES

```
E ostream << DOMString
E XMLFormatter << DOMString
E ostream << DOM_Node
E CMN_PRS_Parser
E ~CMN_PRS_Parser
E UnRepFlags
E ValScheme
E ParseFile
E DumpTree
I ParseNode
I ParseWorkOrder
```

### 9.40.2.Conception Détaillée

Methode : ostream << DOMString

Description : Stream out a DOM string

Parametres :

(ENTREE) target  
(ENTREE) s

Valeur de retour :

ostream

Methode : XMLFormatter << DOMString

Description : Format a DOM string

Parametres :

(ENTREE) strm  
(ENTREE) s

Valeur de retour :

XMLFormatter

-----  
Methode : ostream << DOM\_Node

Description : Stream out a DOM node, and, recursively, all of its children

Parametres :

(ENTREE) target  
(ENTREE) toWrite

Valeur de retour :

ostream

-----  
Methode : CMN\_PRS\_Parser

Description : Constructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Initialise le modèle XML de type XML4C2  
-----

Methode : ~CMN\_PRS\_Parser

Description : Destructeur

Parametres :

Aucun

Valeur de retour :

Aucune

-----  
Nettoyage

Appel de la méthode Terminate  
-----

Methode : UnRepFlags

Description : Met à jour la prise en compte des caractères non visibles

Parametres :

(ENTREE) flag : Mode de gestion à prendre en compte

Valeur de retour :



Aucune

-----  
Methode : ValScheme

Description : Met à jour le schéma de validation

Parametres :

(ENTREE) scheme : Mode de validation à prendre en compte

Valeur de retour :

Aucune

-----  
Methode : ParseFile

Description : Initialise un parser, valide et parse un fichier XML

Parametres :

(ENTREE) xmlFile : Nom du fichier XML lu

Valeur de retour :

CMN\_\_STATUT\_OK si l'arbre XML a été chargé  
CMN\_\_STATUT\_NOK sinon

-----  
Methode : DumpTree

Description : Ecrit un fichier XML à partir d'un arbre XML

Parametres :

(ENTREE) xmlFile : Nom du fichier XML créé

Valeur de retour :

CMN\_\_STATUT\_OK si l'écriture de l'arbre XML a été correcte  
CMN\_\_STATUT\_NOK sinon

-----  
SI un arbre existe  
    SI il n'y a pas eu d'erreurs précédemment  
        Recherche de l'encodage  
        En captant les exceptions  
        Sortie du document  
        Ajouter linefeed dans la sortie  
        Traiter les exceptions  
    FIN SI  
FIN SI

-----  
Methode : ParseNode

Description : Parse un élément d'un arbre de type WorkOrder et met à jour le Work Order

Parametres :

(ENTREE) node  
(ENTREE) element

(SORTIE) workOrder

Valeur de retour :

Aucune

-----  
Methode : ParseWorkOrder

Description : Parse un arbre de type WorkOrder et construit un  
objet Work Order

Parametres :

(SORTIE) workOrder

Valeur de retour :

CMN\_\_STATUT\_OK si l'objet a été construit

CMN\_\_STATUT\_NOK sinon

-----  
SI un arbre lu existe et aucune erreur n'a été détectée  
récupérer le document XML  
parser récursivement l'arbre à partir de la racine  
FIN SI