

CDPP



AKKA Technologies

CDPP-CD-31000-579-AKKA

Edition : 01 Date : 07/02/2017

Révision : 00 Date : 07/02/2017

Réf. : CDPP-CD-31000-579-AKKA

MT : X Code diffusion : E

DOSSIER DE CONCEPTION

CHAINE DE TRAITEMENT DES FOOTPRINTS

Rédigé par : RENARD Benjamin AKKA IS CEPHIRINS Vincent AKKA IS	le : 07/02/2017	
Validé par : LORMANT Nicolas AKKA IS	le : 09/02/2017	
Pour application :	le :	

BORDEREAU D'INDEXATION

CONFIDENTIALITE :

P

DATE DE

DECLASSIFICATION :

MOTS CLES : VALDO, production de données, footprints

TITRE DU DOCUMENT :

DOSSIER DE CONCEPTION

Chaîne de traitement des footprints

AUTEUR(S) :

RENARD Benjamin

AKKA IS

CEPHIRINS Vincent

AKKA IS

RESUME : Ce document présente le dossier de conception de la chaîne de traitement des footprints du CDPP

DOCUMENTS RATTACHES : Ce document vit seul.

LOCALISATION :

VOLUME : 1

NBRE TOTAL DE PAGES : 30

DOCUMENT COMPOSITE : N

LANGUE : FR

DONT PAGES LIMINAIRES : 6

NBRE DE PAGES SUPPL. : 0

GESTION DE CONF. : NG

RESP. GEST. CONF. :

CAUSE D'EVOLUTION : Création du document

CONTRAT : 141666

SYSTEME HOTE :

Microsoft Word 14.0 (14.0.7177)

\\akka.eu\Groupe\PROJETS\BASSO\CNES\ACIS\08_Utilitaires_Outil\GDOC_4312_Freeware\MODEL
ES_GDOC\GDOC_AKKA.dot

Version GDOC : v4.3.1.2

Base projet :

\\akka.eu\groupe\PROJETS\BASSO\CNES\ACIS\08_Utilitaires_Outil\GDOC_4.3.0\bases\VALDO\Vdlib

DIFFUSION INTERNE

Nom	Sigle	Bpi	Observations
DUFOURG Nicolas	DNO/SC/ED	923	

DIFFUSION EXTERNE

Nom	Sigle	Observations
GENOT Vincent	IRAP	
PITOUT Frédéric	IRAP	
BOUCHEMIT Myriam	IRAP	
BUDNIK Elena	IRAP	
GANGLOFF Michel	IRAP	
CEPHIRINS Vincent	AKKA IS	
LORMANT Nicolas	AKKA IS	
RENARD Benjamin	AKKA IS	

MODIFICATION

Ed.	Rév.	Date	Référence, Auteur(s), Causes d'évolution
01	00	07/02/2017	CDPP-CD-31000-579-AKKA RENARD Benjamin AKKA IS CEPHIRINS Vincent AKKA IS Création du document

SOMMAIRE

GLOSSAIRE ET LISTE DES PARAMETRES AC & AD	1
1. GENERALITES	2
1.1. DOCUMENTS DE REFERENCE	2
1.2. DOCUMENTS APPLICABLES	2
2. PRESENTATION GENERALE DE LA CHAINE DE TRAITEMENT	3
2.1. PRESENTATION DES COMPOSANTS.....	3
2.2. WORKFLOW DE PRODUCTION DE LA BASE DE DONNEES DES PARAMETRES DU VENT SOLAIRE	4
2.2.1. Etape de validation des entrées de l'utilisateur.....	4
2.2.2. Etape de récupération des ressources	5
2.2.3. Etape d'extraction et de fusion des données	5
2.2.4. Etape d'interpolation des données	5
2.2.5. Etape de moyennage des données.....	5
2.2.6. Etape de génération d'un fichier générique JSON	5
2.2.7. Etape de génération des fichiers ASCII et des fichiers CDF	6
2.3. WORKFLOW DE PRODUCTION DES BASES DE DONNEES DES FOOTPRINTS	6
2.3.1. Etape de validation des entrées de l'utilisateur.....	7
2.3.2. Etape de validation des paramètres du vent solaire	7
2.3.3. Etape de récupération des orbites.....	7
2.3.4. Etape de conversion du fichier orbites (VOTable) en un fichier générique JSON	7
2.3.5. Etape de génération du fichier CDF des orbites.....	7
2.3.6. Etape de calcul des footprints	7
2.3.7. Etape de conversion des footprints dans le système Quasi dipôle...	7
2.3.8. Etape de génération des fichiers ASCII et des fichiers CDF	8
3. INTERFACES DES PRINCIPAUX COMPOSANTS	9
3.1. COMPOSANT « WRITEJSONDATAINFILE »	9
3.1.1. Paramètre d'entrée	9
3.1.1.1. Description du format de l'objet « file »	9
3.1.1.2. Description du format du fichier « header »	10
3.1.1.3. Description du format de l'objet « Attribute »	10
3.1.1.4. Description du format de l'objet « Column »	10
3.1.1.5. Exemple d'un fichier de commande	11
3.1.1.6. Exemple d'un fichier « header »	11
3.1.1.7. Exemple d'un fichier « records »	12
3.1.2. Données de sortie	12
3.2. COMPOSANT « WS3DVIEWCLIENT »	12

3.3.	COMPOSANT « COMPUTEFOOTPRINTS »	12
3.3.1.	Paramètres d'entrée	12
3.3.1.1.	Description du format de l'objet « data »	13
3.3.1.2.	Description du format de l'objet « positions »	13
3.3.1.3.	Description du format de l'objet « conditions »	14
3.3.1.4.	Description du format de l'objet « output »	14
3.3.1.5.	Exemple d'un fichier d'entrée	15
3.3.2.	Données de sortie	15
3.4.	COMPOSANT « LAUNCHSWPARAMETERS »	16
3.4.1.	Paramètre d'entrée	16
3.4.2.	Données de sortie	16
3.5.	COMPOSANT « LAUNCHFOOTPRINTS »	18
3.5.1.	Paramètres d'entrée	18
3.5.2.	Données de sortie	19

GLOSSAIRE ET LISTE DES PARAMETRES AC & AD

ASCII	American Standard Code for Information Interchange
CDF	Common Data Format
ISO	International Standard Organization

Liste des paramètres AC :

Liste des paramètres AD :

1. GENERALITES

1.1. DOCUMENTS DE REFERENCE

- DR1 Manuel d'installation et d'utilisation - Chaîne de traitement des footprints
RENARD Benjamin, CEPHIRINS Vincent, 07/02/2016, Édit. 01, Rév. 00
CDPP-MU-31000-578-AKKA
- DR2 Interfaces specifications 3DView
BEIGBEDER Laurent, CAUSSARIEU Stéphane, 21/07/2016, Édit. 01, Rév. 00
CDPP-IF-32600-534-GFI

1.2. DOCUMENTS APPLICABLES

- DA1 Spécifications pour la génération de fichiers d'empreintes ionosphériques d'orbites de satellites
PITOUT Frédéric, MARCHAUDON Aurélie, BLELLY Pierre-Louis, 26/10/2016, Édit. 01, Rév. 00
CDPP-ST-31200-574-IRAP
- DA2 CCTP VALDO - Travaux complémentaires 2016 production de données
DUFOURG Nicolas, 27/10/2016, Édit. 1, Rév. 0
CDPP-CT-31200-560-CNES

2. PRESENTATION GENERALE DE LA CHAINE DE TRAITEMENT

La chaîne de traitement des footprints est constituée de deux chaînes produisant des jeux de données sous forme de fichiers :

- Une chaîne de production de la base de données des paramètres du vent solaire,
- Une chaîne de production des bases de données des footprints. Cette chaîne est cliente des données produites à partir de la précédente.

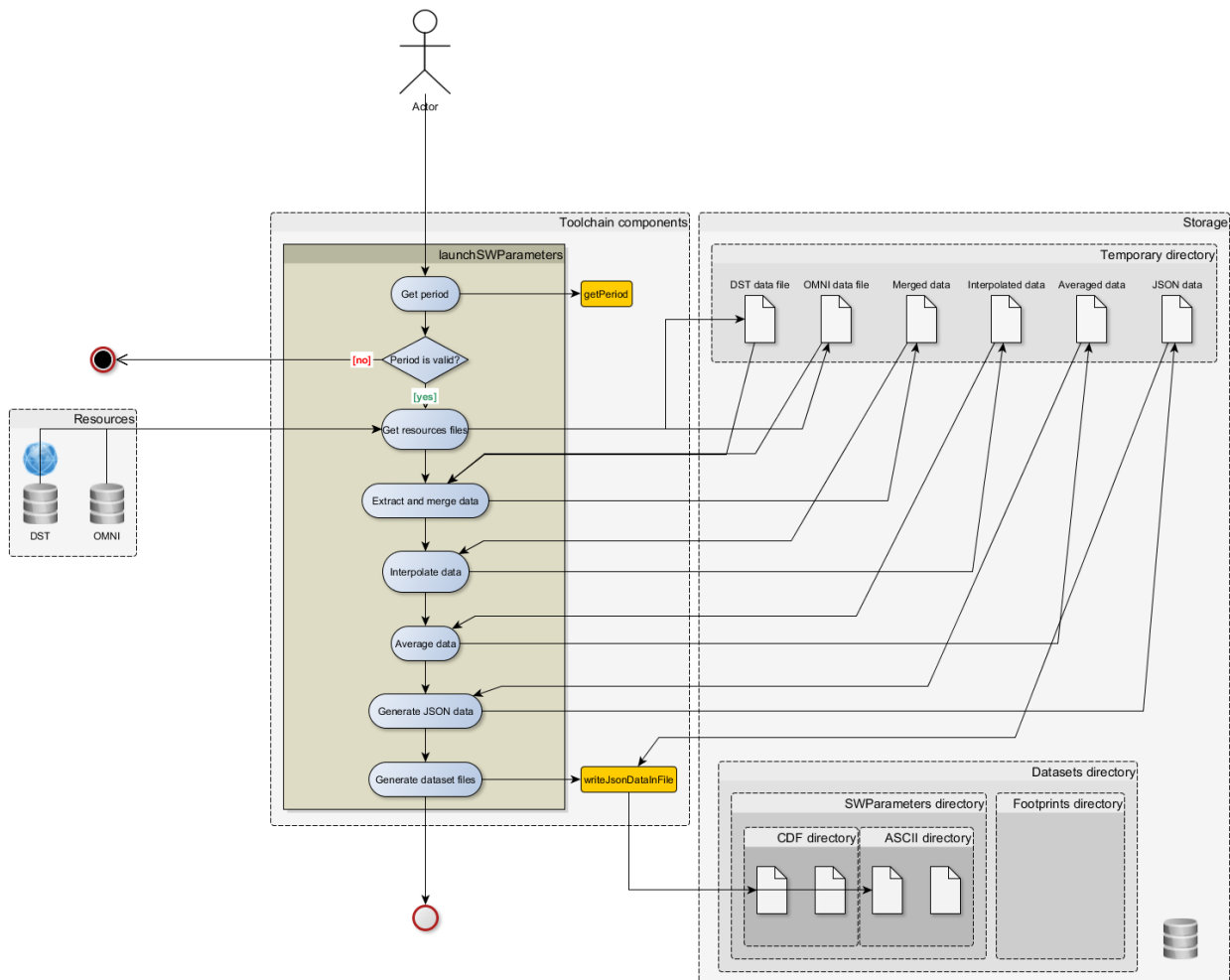
2.1. PRESENTATION DES COMPOSANTS

Les différents composants constituant ces chaînes de traitement sont les suivants :

- Les composants de base :
 - « writeJsonDataInFile » : outil générique permettant de convertir des fichiers de données fournis dans un format JSON (décrit dans §3.1.1) aux formats ASCII et CDF,
 - « getPeriod » : script générique permettant de manipuler les intervalles de temps,
 - « WS3DViewClient » : client SOAP du Webservice de 3DView,
- Les composants spécifiques à la production de la base des paramètres du vent solaire :
 - « launchSWParameters » : script principal de la chaîne de production des paramètres du vent solaire,
- Les composants spécifiques à la production des bases des footprints :
 - « computeFootprints » : outil de calcul des footprints (à partir de l'utilisation de la librairie GEOPAK, fournie en COTS, cf. DR1),
 - « computeFootprintsInQDFrame » : outil de conversion des footprints dans le système de coordonnées géographique vers le système de coordonnées « Quasi-Dipôle »,
 - « getSpacecraftInfo » : script dédié à la récupération des informations relatives à la définition d'une mission au niveau du Webservice de 3DView,
 - « getSpacecraftOrbs » : script dédié à la récupération des orbites d'une mission via l'utilisation du Webservice de 3DView,
 - « convertOrbsFrom3DViewVOTableToJSON » : script de conversion des données d'orbites du format VOTable, fourni par le Webservice de 3DView, vers un format générique JSON (décrit dans §3.1.1),
 - « launchFootprints » : script principal de la chaîne de production des footprints.

2.2. WORKFLOW DE PRODUCTION DE LA BASE DE DONNEES DES PARAMETRES DU VENT SOLAIRE

Le schéma suivant représente les différentes étapes permettant la génération de la base de données des paramètres du vent solaire :



Les sous-sections suivantes décrivent les différentes étapes du composant « launchSWParameters ».

2.2.1. Etape de validation des entrées de l'utilisateur

La définition de l'intervalle de temps sur lequel l'utilisateur souhaite générer la base de données est fournie en entrée de la chaîne de traitement.

Une étape de validation de cette entrée est effectuée, notamment à l'aide de l'utilisation du composant « getPeriod ».

2.2.2. Etape de récupération des ressources

La chaîne de traitement utilise les ressources suivantes :

- Les données OMNI à 1 minute de résolution :
http://spdf.gsfc.nasa.gov/pub/data/omni/high_res_omni/monthly_1min
- Les données du « Geomagnetic Data Service » de Kyoto : <http://wdc.kugi.kyoto-u.ac.jp/dstae/index.html>

Cette étape construit les URL d'accès aux données correspondantes à la période voulue, et les copie dans un espace temporaire.

Dans le cas où ces données sont manquantes, des fichiers contenant des valeurs par défaut sont générés.

2.2.3. Etape d'extraction et de fusion des données

A partir des fichiers OMNI et DST, un fichier ASCII est créé dans l'espace temporaire, contenant les données OMNI voulues (temps, IMF et pression du vent solaire) et la valeur DST la plus proche correspondante.

2.2.4. Etape d'interpolation des données

A cette étape, les données OMNI manquantes de l'étape précédente sont :

- Soit interpolées, si le trou de données est inférieur à 1 heure,
- Soit remplacées par des valeurs par défaut, si le trou de données est supérieur ou égal à 1 heure.

Par ailleurs, le paramètre « quality flag » est renseigné avec les valeurs suivantes :

- 0 : données physiques issues de la ressource OMNI,
- 2 : données interpolées, ou valeurs par défaut.

Le résultat de cette étape est sauvé dans un fichier ASCII, dans l'espace temporaire.

2.2.5. Etape de moyennage des données

Cette étape applique, sur les données de l'étape précédente, une moyenne glissante. Le résultat est sauvé dans un fichier ASCII, dans l'espace temporaire.

2.2.6. Etape de génération d'un fichier générique JSON

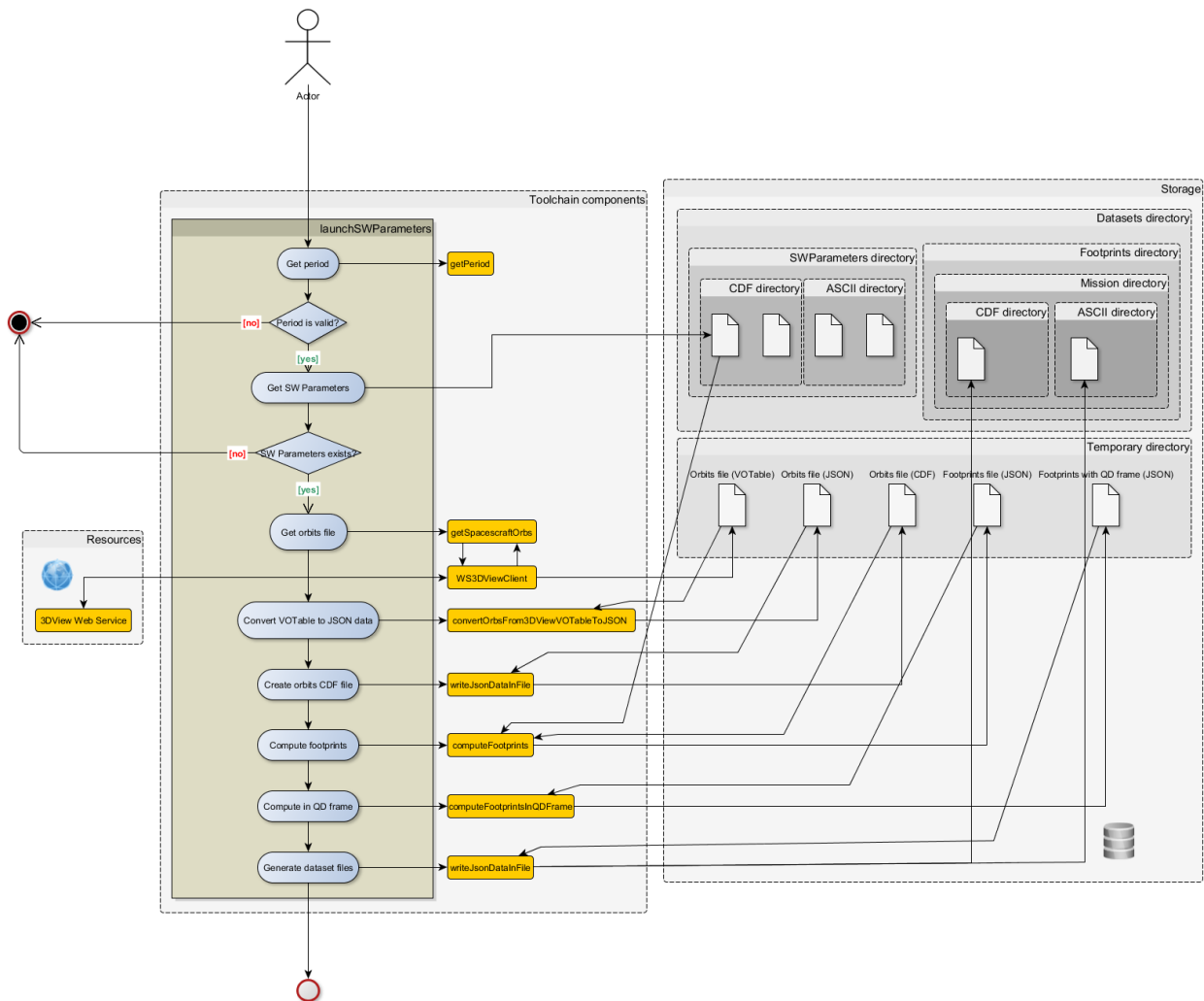
Cette étape convertit le fichier ASCII de l'étape précédente en un fichier de données « JSON », dont le format est décrit dans §3.1.1.

2.2.7. Etape de génération des fichiers ASCII et des fichiers CDF

Cette étape utilise le composant « writeJsonDataInFile » pour produire, à partir du fichier JSON de l'étape précédente, les fichiers finaux aux formats ASCII et CDF.

2.3. WORKFLOW DE PRODUCTION DES BASES DE DONNEES DES FOOTPRINTS

Le schéma suivant représente les différentes étapes permettant la génération de la base de données des footprints pour une mission donnée :



Les sous-sections suivantes décrivent les différentes étapes du composant « launchFootprints ».

2.3.1. Etape de validation des entrées de l'utilisateur

L'utilisateur fournit :

- Le nom de la mission pour laquelle il souhaite générer les footprints,
- La période sur laquelle appliquer la génération.

Ces entrées sont validées à cette étape, notamment à l'aide de l'utilisation du composant « getPeriod » et du composant « getSpacecraftInfo » (non représenté dans le schéma précédent, par soucis de clarté).

2.3.2. Etape de validation des paramètres du vent solaire

Pour la période demandée, le fichier des paramètres du vent solaire (cf. §2.2) correspondant, au format CDF, doit exister.

Cette étape teste donc l'existence de ce fichier.

2.3.3. Etape de récupération des orbites

Cette étape utilise le composant « getSpacecraftOrbs », qui utilise lui-même le composant « WS3DViewClient » afin de récupérer le fichier VOTable des orbites correspondantes.

2.3.4. Etape de conversion du fichier orbites (VOTable) en un fichier générique JSON

A cet étape, le fichier VOTable de l'étape précédente est converti en un fichier de données JSON, dont le format est décrit dans §3.1.1.

2.3.5. Etape de génération du fichier CDF des orbites

A partir du fichier de données JSON de l'étape précédente, un fichier CDF des orbites est généré via l'utilisation du composant « writeJsonDataInFile ».

2.3.6. Etape de calcul des footprints

A partir du fichier CDF des orbites généré à l'étape précédente, et du fichier des paramètres du vent solaire (cf. §2.3.2), le composant « computeFootprints » calcule les footprints dans le système de coordonnées géographique.

Le résultat de ce calcul est écrit dans un fichier de données JSON, dont le format est décrit dans §3.1.1.

2.3.7. Etape de conversion des footprints dans le système Quasi dipôle

A cette étape, le fichier de données JSON de l'étape précédente est enrichi avec l'écriture des footprints dans le système de coordonnées « quasi dipôle ».

2.3.8. Etape de génération des fichiers ASCII et des fichiers CDF

A partir du fichier de données JSON de l'étape précédente, les fichiers ASCII et CDF de la base des footprints sont générés via l'utilisation du composant « writeJsonDataInFile ».

3. INTERFACES DES PRINCIPAUX COMPOSANTS

3.1. COMPOSANT « WRITEJSONDATAINFILE »

3.1.1. Paramètre d'entrée

Le paramètre d'entrée de ce composant est un fichier JSON respectant le format suivant :

Properties	Mandatory ?	Type	Description
command	Yes	String	Type de la commande à appliquer (actuellement « create » ou « push »).
file	Yes	Object (format décrit dans §3.1.1.1).	Informations concernant le fichier de sortie.
header	Yes	String	Chemin absolu vers le fichier « header » (format JSON décrit dans §3.1.1.2).
records	Yes	String	Chemin absolu vers le fichier des données (organisées sous forme d'un tableau des enregistrements, contenant eux même un tableau des valeurs correspondantes).

3.1.1.1. Description du format de l'objet « file »

Properties	Mandatory ?	Type	Description
name	Yes	String	Nom du fichier à générer.
path	Yes	String	Chemin absolu du répertoire de destination
format	Yes	String	Format du fichier à générer (actuellement ASCII ou CDF).

3.1.1.2. Description du format du fichier « header »

Properties	Mandatory ?	Type	Description
attributes	Yes	Array of Attribute objects	Liste des attributs globaux (format de l'objet Attribute décrit dans §3.1.1.3).
columns	Yes	Array of Column objects	Liste des attributs globaux (format de l'objet Column décrit dans §3.1.1.4).

3.1.1.3. Description du format de l'objet « Attribute »

Properties	Mandatory ?	Type	Description
name	Yes	String	Nom de l'attribut
value	Yes	String	Valeur de l'attribut

3.1.1.4. Description du format de l'objet « Column »

Properties	Mandatory ?	Type	Description
name	Yes	String	Nom de la colonne
type	Yes	String	Type de la colonne (actuellement : « time », « float » ou « int »).
format	Only if type is « time »	String	Description du format du temps (actuellement, seul le format « YYYY-MM-DDThh:mm:ssZ » est disponible)
attributes	Yes	Array of Attribute objects	Liste des attributs spécifiques à cette colonne (format de l'objet Attribute décrit dans §3.1.1.3).

3.1.1.5. Exemple d'un fichier de commande

```
{
  "command" : "create",
  "file" : {
    "name" : "orbitSpacecraft_CLUSTER1_20100101_20100131.cdf",
    "path" : "/tmp/dev/footprints/data",
    "format" : "cdf"
  },
  "header": "/tmp/dev/footprints/data/orbitSpacecraft_header_CLUSTER1_20100101_20100131.json",
  "records": "/tmp/dev/footprints/data/orbitSpacecraft_data_CLUSTER1_20100101_20100131.json"
}
```

3.1.1.6. Exemple d'un fichier « header »

```
{
  "attributes": [{
    "name": "Generator",
    "value": "CDPP/Footprints generator v1.0"
  }, {
    "name": "Generation date",
    "value": "2017-01-23T12:07:36Z"
  }],
  "columns": [{
    "name": "time",
    "type": "time",
    "format": "YYYY-MM-DDThh:mm:ssZ",
    "attributes": [{
      "name": "description",
      "value": "Time in ISO 8601"
    }]
  }, {
    "name": "xyz_GSM",
    "type": "float",
    "dims": [3],
    "attributes": [{
      "name": "description",
      "value": "Input orbit"
    }], {
    "name": "labels",
```

```

        "value": "X, Y, Z"
    }, {
        "name": "units",
        "value": "Re"
    }, {
        "name": "syscoord",
        "value": "GSM"
    }
  ]
}

```

3.1.1.7. Exemple d'un fichier « records »

```

[
  ["2010-01-01T00:00:00Z", "9.04589", "12.7136", "-6.52672"],
  ["2010-01-01T00:01:00Z", "9.04521", "12.6994", "-6.52787"]
]

```

3.1.2. Données de sortie

Fichier ASCII ou CDF correspondant.

3.2. COMPOSANT « WS3DVIEWCLIENT »

Ce composant est un client SOAP du Webservice de 3DView dont les interfaces sont définies dans DR2 .

3.3. COMPOSANT « COMPUTEFOOTPRINTS »

3.3.1. Paramètres d'entrée

Le paramètre d'entrée de ce composant est un fichier JSON respectant le format suivant :

Properties	Mandatory ?	Type	Description
data	Yes	Object (format JSON)	Objet décrivant les données

		décrit dans §3.3.1.1)	d'entrée utilisées pour le calcul
output	Yes	Object (format JSON décrit dans §3.3.1.4).	Informations concernant les fichiers de sortie.

3.3.1.1. Description du format de l'objet « data »

Properties	Mandatory ?	Type	Description
positions	Yes	Object (format JSON décrit dans §3.3.1.2)	Objet décrivant les données d'entrée d'orbites
conditions	Yes	Object (format JSON décrit 3.3.1.3)	Objet décrivant les données d'entrée des paramètres du vent solaire
inputHeaderFilePath	Yes	String	Chemin vers le fichier « header » d'entrée (cf. §3.1.1.2)

3.3.1.2. Description du format de l'objet « positions »

Properties	Mandatory ?	Type	Description
filePath	Yes	String	Chemin absolu vers le fichier CDF des orbites
timeParamName	Yes	String	Nom du paramètre du temps
posParamName	Yes	String	Nom du paramètre de position
attributes	Yes	Array of Strings	Liste des attributs globaux du fichier des orbites à copier dans le fichier de sortie

3.3.1.3. Description du format de l'objet « conditions »

Properties	Mandatory ?	Type	Description
filePath	Yes	String	Chemin absolu vers le fichier CDF des paramètres du vent solaire
timeParamName	Yes	String	Nom du paramètre du temps
dstParamName	Yes	String	Nom du paramètre DST
imfParamName	Yes	String	Nom du paramètre IMF
pswParamName	Yes	String	Nom du paramètre de pression du vent solaire
qualityFlagParamName	Yes	String	Nom du paramètre de l'indice de qualité
attributes	Yes	Array of Strings	Liste des attributs globaux du fichier des orbites à copier dans le fichier de sortie

3.3.1.4. Description du format de l'objet « output »

Properties	Mandatory ?	Type	Description
dataFilePath	Yes	String	Chemin absolu vers le fichier de sortie contenant les données des footprints
headerFilePath	Yes	String	Chemin absolu vers le fichier de sortie contenant la description de l'entête des données des footprints

3.3.1.5. Exemple d'un fichier d'entrée

```
{
  "data" : {
    "positions" : {
      "filePath" : "/tmp/dev/footprints/data/orbitSpacecraft_GEOTAIL_20100101_20100131.cdf",
      "timeParamName" : "time",
      "posParamName" : "xyz_gsm",
      "attributes" : [
        "Orbit source",
        "Orbit kernel name",
        "Orbit kernel version"
      ]
    },
    "conditions" : {
      "filePath" : "/home/dev/footprints/./datasets/sw_parameters/cdf/sw_parameters_201001.cdf",
      "timeParamName" : "time",
      "dstParamName" : "Dst",
      "imfParamName" : "IMF",
      "pswParamName" : "Psw",
      "qualityFlagParamName" : "quality_flag",
      "attributes" : [
        "OMNI input URL",
        "DST input URL"
      ]
    },
    "inputHeaderFilePath" : "/home/dev/footprints/./config/templates/header_footprints.json.tpl"
  },
  "output" : {
    "dataFilePath" : "/tmp/dev/footprints/data/footprints_data_GEOTAIL_20100101_20100131.json",
    "headerFilePath" :
"/tmp/dev/footprints/data/footprints_header_GEOTAIL_20100101_20100131.json.tpl"
  }
}
```

3.3.2. Données de sortie

Les données de sorties sont deux fichiers JSON :

- Un fichier contenant les données « records » correspondantes (cf. §3.1.1.7),
- Un fichier contenant la description de l'entête du fichier (cf. §3.1.1.6).

3.4. COMPOSANT « LAUNCHSWPARAMETERS »

3.4.1. Paramètre d'entrée

Le paramètre d'entrée de ce composant est la période de génération de la base de données des paramètres du vent solaire (cf. DR1).

3.4.2. Données de sortie

Les données de sortie de ce composant sont des fichiers ASCII et CDF, dont la structure respecte le template suivant (cf. §3.1.1.2) :

```
{
  "attributes": [
    {
      "name": "Generator",
      "value": "##TAG_GENERATOR##"
    },
    {
      "name" : "Generation date",
      "value" : "##TAG_GENERATION_DATE##"
    },
    {
      "name" : "Start date",
      "value" : "##TAG_START_DATE##"
    },
    {
      "name" : "Stop date",
      "value" : "##TAG_STOP_DATE##"
    },
    {
      "name" : "OMNI input URL",
      "value" : "##TAG_OMNI_FILE##"
    },
    {
      "name" : "DST input URL",
      "value" : "##TAG_DST_FILE##"
    }
  ],
  "columns": [
    {
      "name": "time",
      "type": "time",
    }
  ]
}
```

```
"format": "YYYY-MM-DDThh:mm:ssZ",
"attributes": [
  {
    "name": "description",
    "value": "Time in ISO 8601"
  }
]
},
{
  "name" : "IMF",
  "type" : "float",
  "dims" : [3],
  "attributes" : [
    {
      "name": "description",
      "value": "Input IMF data"
    },
    {
      "name" : "labels",
      "value" : "Bx, By, Bz"
    },
    {
      "name" : "units",
      "value" : "nT"
    },
    {
      "name" : "syscoord",
      "value" : "GSM"
    }
  ]
},
{
  "name" : "Psw",
  "type" : "float",
  "attributes" : [
    {
      "name": "description",
      "value": "Flow pressure"
    },
    {
      "name": "units",
      "value": "nPa"
    }
  ]
}
```

```
  },
  {
    "name" : "Dst",
    "type" : "float",
    "attributes" : [
      {
        "name": "description",
        "value": "Dst indice"
      },
      {
        "name" : "units",
        "value" : "nT"
      }
    ]
  },
  {
    "name" : "quality_flag",
    "type" : "int",
    "attributes" : [
      {
        "name" : "description",
        "value" : "Quality flag related to OMNI data"
      },
      {
        "name" : "flags",
        "value" : "0 : optimal, 2 : interpolated or default values"
      }
    ]
  }
]
```

Note : Les tags écrits sous la forme « ##TAG_****## » sont remplacés automatiquement par les valeurs correspondantes lors de l'exécution du composant.

3.5. COMPOSANT « LAUNCHFOOTPRINTS »

3.5.1. Paramètres d'entrée

Les paramètres d'entrée de ce composant sont le nom du satellite et la période de génération de la base de données des footprints (cf. DR1).

3.5.2. Données de sortie

Les données de sortie de ce composant sont des fichiers ASCII et CDF, dont la structure respecte le template suivant (cf. §3.1.1.2) :

```
{
  "attributes" : [
    {
      "name" : "Generator",
      "value" : "##TAG_GENERATOR##"
    },
    {
      "name" : "Generation date",
      "value" : "##TAG_GENERATION_DATE##"
    },
    {
      "name" : "Satellite name",
      "value" : "##TAG_SATELLITE_NAME##"
    },
    {
      "name" : "Start date",
      "value" : "##TAG_START_DATE##"
    },
    {
      "name" : "Stop date",
      "value" : "##TAG_STOP_DATE##"
    }
  ]
  ##TAG_INPUT_ATTRIBUTES##
},
"columns" : [
  {
    "name" : "time",
    "type" : "time",
    "format" : "YYYY-MM-DDThh:mm:ssZ",
    "attributes" : [
      {
        "name": "description",
        "value": "Time in ISO 8601"
      }
    ]
  }
],
{
  "name" : "xyz_GSM",
  "type" : "float",
```

```
"dims"      : [3],
"attributes" : [
  {
    "name": "description",
    "value": "Input orbit"
  },
  {
    "name" : "labels",
    "value" : "X, Y, Z"
  },
  {
    "name" : "units",
    "value" : "Re"
  },
  {
    "name" : "syscoord",
    "value" : "GSM"
  }
]
},
{
  "name" : "IMF",
  "type"  : "float",
  "dims"  : [3],
  "attributes" : [
    {
      "name": "description",
      "value": "Input IMF data"
    },
    {
      "name" : "labels",
      "value" : "Bx, By, Bz"
    },
    {
      "name" : "units",
      "value" : "nT"
    },
    {
      "name" : "syscoord",
      "value" : "GSM"
    }
  ]
},
{
```

```
"name" : "Psw",
"type"  : "float",
"attributes" : [
  {
    "name": "description",
    "value": "Input dynamic pressure"
  },
  {
    "name": "units",
    "value": "nPa"
  }
]
},
{
  "name" : "Dst",
  "type"  : "float",
  "attributes" : [
    {
      "name": "description",
      "value": "Dst Indice"
    },
    {
      "name" : "units",
      "value" : "nT"
    }
  ]
},
{
  "name" : "quality_flag",
  "type"  : "int",
  "attributes" : [
    {
      "name" : "description",
      "value" : "0 : optimal, 1: inside magnetosphere but distant, 2 : OMNI data interpolated, 3 :
outside magnetosphere"
    }
  ]
},
{
  "name" : "footprint_north_geo",
  "type" : "float",
  "dims" : [2],
  "attributes" : [
    {
```

```
    "name": "description",
    "value": "Footprint north"
  },
  {
    "name" : "labels",
    "value" : "lat, lon"
  },
  {
    "name" : "units",
    "value" : "deg"
  },
  {
    "name" : "syscoord",
    "value" : "geographic"
  }
]
},
{
  "name" : "footprint_south_geo",
  "type" : "float",
  "dims" : [2],
  "attributes" : [
    {
      "name": "description",
      "value": "Footprint south"
    },
    {
      "name" : "labels",
      "value" : "lat, lon"
    },
    {
      "name" : "units",
      "value" : "deg"
    },
    {
      "name" : "syscoord",
      "value" : "geographic"
    }
  ]
}
},
{
  "name" : "footprint_north_qd",
  "type" : "float",
  "dims" : [3],
```

```
"attributes" : [
  {
    "name": "description",
    "value": "Footprint north"
  },
  {
    "name" : "labels",
    "value" : "lat, lon, MLT"
  },
  {
    "name" : "units",
    "value" : "deg, deg, hour"
  },
  {
    "name" : "syscoord",
    "value" : "Quasi-Dipole"
  }
]
},
{
  "name" : "footprint_south_qd",
  "type" : "float",
  "dims" : [3],
  "attributes" : [
    {
      "name": "description",
      "value": "Footprint south"
    },
    {
      "name" : "labels",
      "value" : "lat, lon, MLT"
    },
    {
      "name" : "units",
      "value" : "deg, deg, hour"
    },
    {
      "name" : "syscoord",
      "value" : "Quasi-Dipole"
    }
  ]
}
]
```

AKKA IS

CDPP

CDPP-CD-31000-579-AKKA

Edit. : 01

Date : 07/02/2017

Rév. : 00

Date : 07/02/2017

Référence : [CDPP-CD-31000-579-AKKA](#)

Page : 24

Note : Les tags écrits sous la forme « ##TAG_***## » sont remplacés automatiquement par les valeurs correspondantes lors de l'exécution du composant.