

IASI

THALES INFORMATION SYSTEMS

IA-PE-2100-9559-THA


Edition : 03 Date : 21/02/2003

Révision : 08 Date : 28/07/2009

MT : X Code diffusion : E

Réf. : -

**PLAN D'ESSAI DE VALIDATION SYSTEME / PLAN D'ESSAI,  
D'INTEGRATION, DE QUALIFICATI  
PLAN D'ESSAIS DU LOGICIEL OPS IASI**

<b>Rédigé par :</b> BRUNEL Samuel	THALES SERVICES	le : 28/07/2009	
<b>Validé par :</b> RANDRIA Prosper	THALES SERVICES	le : 28/07/2009	p.o. 
<b>Pour application :</b> BOTELLA Christine	THALES SERVICES	le : 28/07/2009	

**BORDEREAU D'INDEXATION****CONFIDENTIALITE :**  
NC**MOTS CLES :** Essai, Validation, IASI, Traitement d'images**TITRE DU DOCUMENT :** PLAN D'ESSAI DE VALIDATION SYSTEME / PLAN D'ESSAI, D'INTEGRATION, DE QUALIFICATI**PLAN D'ESSAIS DU LOGICIEL OPS IASI****AUTEUR(S) :** BRUNEL Samuel

THALES SERVICES

**RESUME :** Ce document a pour buts de définir une stratégie d'essais et les procédures de tests nécessaires pour démontrer que le logiciel OPS-IASI est conforme à ses spécifications techniques.**DOCUMENTS RATTACHES :** Ce document vit seul.**LOCALISATION :****VOLUME :** 1**NBRE TOTAL DE PAGES :** 285**DONT PAGES LIMINAIRES :** 6**NBRE DE PAGES SUPPL. :** 0**DOCUMENT COMPOSITE :** N**LANGUE :** FR**GESTION DE CONF. :** F**RESP. GEST. CONF. :** BOTELLA**CAUSE D'EVOLUTION :** Livraison OPS V5-0 (DAY2)**CONTRAT :** 01/8937**SYSTEME HOTE :**

Microsoft Word 11.0 (11.0.6568)

G:\Thales\prive\Projets\iasi-ops\Modèles CNES\GDOC 4.0.5\ModeleGDOCIndus.dot

Version GDOC : v4.0.5

Base projet : G:\Thales\prive\Projets\iasi-ops\Modèles CNES\IASI-thales01

## DIFFUSION INTERNE

Nom	Sigle	BPi	Observations
BLUMSTEIN Denis	DCT/PO/EV	2504	
CHALON Gilles	DCT/PO/EV	2504	
PONCE Ghislaine	DCT/PO/EV	2504	
SEGALEN Barbara	DCT/PO/EV	2504	
MARQUIER Henry	DCT/PO/EV	1321	
MORENO Richard	DCT/PS/TIS	1321	
BAILLY Isabelle	DCT/PS/TIS	1321	
RAYSSIGUIER Michel	DCT/PS/TIS	1321	
RICHARD Pascal	DSI/EP/SL	3517	
GLEYZES Jean-Pierre	DCT/PS/TIS	1321	
GOMEZ Marie-Hélène	DCT/PS/TIS	1321	
MATHIEU Nathalie	EUROGICIEL	1415	
LONJOU Vincent	DCT/ME/EI		

## DIFFUSION EXTERNE

Nom	Sigle	Observations
PASCAL Jean-Luc	THALES SERVICES	
MERIGUET Christelle	THALES SERVICES	
BOTELLA Christine	THALES SERVICES	
MASSART Benjamin	THALES SERVICES	
BRUNEL Samuel	THALES SERVICES	
RANDRIA Prosper	THALES SERVICES	

## MODIFICATION

Ed.	Rév.	Date	Référence, Auteur(s), Causes d'évolution
03	08	28/07/2009	- BRUNEL Samuel THALES SERVICES Livraison OPS V5-0 (DAY2)
03	07	20/11/2008	- BRUNEL Samuel THALES SERVICES DM2458 : Migration vers AIX 6.1 et compilateur XLC v 8
03	06	06/11/2006	- PASCAL Jean-Luc THALES SERVICES MERIGUET Christelle THALES SERVICES Mise a jour pour la recette V3.6
03	05	30/04/2004	- PASCAL Jean-Luc THALES IS Mise a jour pour la recette V3.0
03	04	01/10/2003	- PASCAL Jean-Luc THALES IS Mise a jour suite à la recette V2.0
03	03	17/09/2003	- PASCAL Jean-Luc THALES IS Mise a jour pour la recette V2.0
03	02	05/06/2003	- PASCAL Jean-Luc THALES IS Mise a jour suite à la recette V1
03	01	21/05/2003	- BRANET Pascal THALES IS Mise a jour pour la recette V1 et prise en compte de l'action 222
03	00	21/02/2003	- BRANET Pascal THALES IS Prise en compte de l'action A190 (remarques émises lors des recettes V0-0 et V0-1). Insertion des Procédures d'Essais V1 et V2
02	02	29/11/2002	- BRANET Pascal THALES IS Livraison des procédures de validation de la V0 en version définitive Prise en compte IA-IN-2100-9535-CNE
02	01	13/11/2002	- BRANET Pascal THALES IS Livraison des procédures de validation de la V0 en version Draft
02	00	23/08/2002	- BRANET Pascal THALES IS

			Mise au format GDOC, mise à jour suite aux remarques de la RCP et des évolution des spécifications logicielles [DA104]
01	00	14/06/2002	- BRANET Pascal Création du document
			THALES IS

## SOMMAIRE

<b>GLOSSAIRE ET LISTE DES PARAMETRES AC &amp; AD .....</b>	<b>5</b>
<b>1. GENERALITES .....</b>	<b>7</b>
<b>1.1. DOCUMENTS DE REFERENCE ET APPLICABLES .....</b>	<b>7</b>
<b>2. INTRODUCTION .....</b>	<b>8</b>
<b>2.1. OBJECTIF .....</b>	<b>8</b>
<b>2.2. DOMAINE D'APPLICATION .....</b>	<b>9</b>
<b>2.3. PRESENTATION GENERALE .....</b>	<b>9</b>
<b>2.3.1. Contexte .....</b>	<b>9</b>
<b>2.3.2. Diagramme de Contexte .....</b>	<b>11</b>
<b>2.3.3. Rappel des fonctionnalités .....</b>	<b>12</b>
<b>3. PREVISION DE TESTS .....</b>	<b>13</b>
<b>3.1. BUT DE CHAQUE PHASE .....</b>	<b>13</b>
<b>3.2. STRATEGIE DE TEST .....</b>	<b>13</b>
<b>3.2.1. Identification des Interfaces Internes .....</b>	<b>13</b>
<b>3.2.2. Stratégie d'Intégration .....</b>	<b>13</b>
3.2.2.1. Intégration du Serveur de Données .....	14
3.2.2.2. Intégration Système de l'OPS .....	15
<b>3.2.3. Stratégie de Validation .....</b>	<b>16</b>
<b>3.3. ORGANISATION ET PLANIFICATION .....</b>	<b>18</b>
<b>3.4. RESSOURCES HUMAINES ET RESPONSABILITES .....</b>	<b>18</b>
<b>3.5. RESSOURCES MATERIELLES ET LOGICIELLES .....</b>	<b>19</b>
<b>3.5.1. Ressources matérielles .....</b>	<b>19</b>
<b>3.5.2. Jeux de Données .....</b>	<b>19</b>
3.5.2.1. Jeux de Données V0 – V1 .....	20
3.5.2.2. Jeux de Données V2 .....	29
<b>3.5.3. Ressources Logicielles .....</b>	<b>31</b>
<b>3.5.4. Environnement de la Validation .....</b>	<b>32</b>
<b>3.5.5. Outils de validation .....</b>	<b>35</b>
3.5.5.1. Outils de Génération de Données .....	35
3.5.5.2. Outils de Vérification de Données .....	36
<b>4. DESCRIPTION DES TESTS .....</b>	<b>38</b>
<b>4.1. PRESENTATION DES TESTS .....</b>	<b>38</b>
<b>4.2. FORMALISME DES TESTS .....</b>	<b>38</b>
<b>4.3. MODE DE COUVERTURE DES EXIGENCES .....</b>	<b>40</b>
<b>4.3.1. Exigences validées par Inspection .....</b>	<b>40</b>
4.3.1.1. Inspection de la Conception .....	40
4.3.1.2. Inspection du Logiciel .....	41
4.3.1.3. Exigences de Développement .....	41
4.3.1.4. Analyse des Résultats des Tests de validation .....	41
<b>4.3.2. Exigences non Validées .....</b>	<b>42</b>
4.3.2.1. Exigences Non Applicables .....	42
4.3.2.2. Non Implémentées par le PGE .....	43
4.3.2.3. Non Conformes .....	43
4.3.2.4. Non Testables .....	43

<b>4.3.3. Classe d'Essais BUILD</b>	<b>43</b>
4.3.3.1. GEN : Production du Logiciel	43
4.3.3.2. INST : Installation du Logiciel	43
<b>4.3.4. Classe d'Essais FUNCT</b>	<b>44</b>
4.3.4.1. RUN : Lancement de l'OPS	44
4.3.4.2. CONF : Configuration de l'OPS	44
4.3.4.3. GRAN : Traitement de granules en mode Nominal	44
4.3.4.4. GRAI : Traitement de granules en mode Intermédiaire	44
4.3.4.5. DUMP : Traitement d'un Dump Complet	45
4.3.4.6. OVER : Gestion de l'Overlap	45
4.3.4.7. COLD : Traitement nominal avec initialisation à froid	45
4.3.4.8. LOG : Le Journal de Bord	45
4.3.4.9. ERR : Arrêt sur cas dégradé	45
<b>4.3.5. Classe d'Essais M&amp;C</b>	<b>45</b>
4.3.5.1. CMD : Traitement des Commandes Stop/Break/Suspend/Resume/Abort	45
4.3.5.2. HKTM : Génération de la HKTM-Status	45
<b>4.3.6. Classe d'Essais PERF</b>	<b>45</b>
<b>4.3.7. Classe d'Essais ALGO</b>	<b>46</b>
<b>4.3.8. Classe d'Essais OPER</b>	<b>46</b>
4.3.8.1. DBG: Fonctionnement en mode DEBUG	46
4.3.8.2. STD : Fonctionnement en Standalone	46
4.3.8.3. PROC: Fonctionnement sur 2 processeurs	46
4.3.8.4. MULT-Exécution de plusieurs instances	46
<b>5. PROCEDURES ET JEUX D'ESSAIS</b>	<b>47</b>
<b>5.1. TESTS D'INTEGRATION</b>	<b>47</b>
5.1.1. INT-ALGO-N-01	47
5.1.2. INT-SD-N-01	49
5.1.3. INT-STRUC-N-01	51
5.1.4. INT-STRUC-D-02	52
5.1.5. INT-APPL-N-01	53
5.1.6. INT_APPL_N02	55
5.1.7. INT-OPS-N-01	57
<b>5.2. TESTS DE VALIDATION</b>	<b>58</b>
5.2.1. GEN-OPS-N-01	58
5.2.2. INST-OPS-N-01	60
5.2.3. INST-OPS-N-02	62
5.2.4. INST-OPS-N-03	64
5.2.5. SIM-OPS-N-01	66
5.2.6. SIM-OPS-N-02	68
5.2.7. RUN-OPS-N-01	70
5.2.8. RUN-OPS-D-02	72
5.2.9. RUN-OPS-D-03	73
5.2.10. CONF-OPS-N-01	74
5.2.11. CONF-OPS-D-02	76
5.2.12. CONF-OPS-D-03	79
5.2.13. CONF-OPS-D-04	82
5.2.14. CONF-OPS-D-05	85
5.2.15. CONF-OPS-D-06	87
5.2.16. GRAN-OPS-D-02	89
5.2.17. GRAN-OPS-N-04	91
5.2.18. GRAN-OPS-N-05	94
5.2.19. GRAN-OPS-N-06	96



5.2.20. GRAN-OPS-N-07 .....	99
5.2.21. GRAN-OPS-N-08 .....	102
5.2.22. GRAN-OPS-N-09 .....	105
5.2.23. GRAN-OPS-D-10 .....	107
5.2.24. GRAN-OPS-D-11 .....	122
5.2.25. GRAN-OPS-D-12 .....	125
5.2.26. GRAN-OPS-D-13 .....	128
5.2.27. GRAN-OPS-D-14 .....	130
5.2.28. GRAN-OPS-D-15 .....	133
5.2.29. GRAN-OPS-N-16 .....	135
5.2.30. GRAN-OPS-N-17 CORREG .....	138
5.2.31. GRAN-OPS-N-18 .....	141
5.2.32. GRAN-OPS-N-19 .....	143
5.2.33. GRAI-OPS-N-01 .....	146
5.2.34. GRAI-OPS-N-02 .....	150
5.2.35. DUMP-OPS-N-01 .....	155
5.2.36. OVER-OPS-N-01 .....	157
5.2.37. OVER-OPS-D-02 .....	158
5.2.38. COLD-OPS-N-01 .....	161
5.2.39. LOG-OPS-N-01 .....	164
5.2.40. ERR-OPS-N-01 .....	166
5.2.41. CMD-OPS-N-01 .....	167
5.2.42. HKTM-OPS-N-01 .....	171
5.2.43. PERF-OPS-N-01 .....	172
5.2.44. PERF-OPS-N-02 .....	174
5.2.45. ALGO-OPS-N-01 .....	177
5.2.46. ALGO-OPS-N-02 .....	180
5.2.47. ALGO-OPS-N-03 .....	184
5.2.48. ALGO-OPS-N-04 .....	187
5.2.49. ALGO-OPS-N-05 .....	192
5.2.50. ALGO-OPS-N-06 .....	194
5.2.51. ALGO-OPS-N-07 .....	196
5.2.52. ALGO-OPS-N-08 .....	199
5.2.53. ALGO-OPS-N-09 .....	212
5.2.54. ALGO-OPS-N-10 .....	219
5.2.55. ALGO-OPS-N-11 .....	222
5.2.56. ALGO-OPS-N-12 .....	226
5.2.57. ALGO-OPS-N-13 .....	230
5.2.58. ALGO-OPS-N-14 .....	233
5.2.1. ALGO-OPS-N-15 .....	236
5.2.2. ALGO-OPS-N-16 .....	240
5.2.3. ALGO-OPS-N-17 .....	246
5.2.4. ALGO-OPS-N-18 .....	249
5.2.5. ALGO-OPS-N-19 .....	252
5.2.6. DBG-OPS-N-01 .....	255
5.2.7. DBG-OPS-N-02 .....	256
5.2.8. DBG-OPS-D-03 .....	257
5.2.9. STD-OPS-N-01 .....	258
5.2.10. PROC-OPS-N-01 .....	260
5.2.11. MULTI-OPS-N-01 .....	262

ANNEXE A :FICHE TEST .....	A.263
----------------------------	-------

ANNEXE B :FICHE JOURNAL D'ESSAI .....	B.265
---------------------------------------	-------

**ANNEXE C :MATRICE DE COUVERTURE DES INTERFACES..... C.267**

**ANNEXE D :MATRICE DE COUVERTURE DES EXIGENCES..... D.270**

**ANNEXE A : FICHE TEST ..... A.192**

**ANNEXE B : FICHE JOURNAL D'ESSAI ..... B.194**

**ANNEXE C : MATRICE DE COUVERTURE DES INTERFACES..... C.196**

**ANNEXE D : MATRICE DE COUVERTURE DES EXIGENCES..... D.199**

## GLOSSAIRE ET LISTE DES PARAMETRES AC & AD

AC	A Confirmer
AD	A Définir
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
DDC	Dossier Descriptif de Configuration
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
EUMETSAT	EUMETSAT est une organisation intergouvernementale regroupant 17 nations européenne, dont l'objectif est l'établissement, le maintien et l'exploitation des systèmes européens de satellites météorologiques opérationnels
HRPT	High Resolution Picture Transmission : centres de traitements locaux qui exploitent les données METOP qui descendent en bande L.
IASI	Infrared Atmospheric Sounding Interferometer : interféromètre de sondage atmosphérique dans l'infrarouge.
ISRFEM	Instrument Spectral Response Function Estimation Model
JDBS	JdB server
MCS	Monitoring and Control Segment
METOP	Série de satellites météorologiques opérationnels en orbite polaire. IASI est l'un des instruments de METOP.
MLA	MCS Local Agent
MNP	Modèle Numérique de Performance.
MP	Main Process
MSGs	Message Server
OPS	Logiciel Opérationnel (Operational Software) : correspond au IASI level 1 PPS dans les glossaires d'EUMETSAT. PPS=Product Processing Software
PGE	Product Generation Element : fournit des services aux PPS [DA1]
PGF	Product Generation Facility
PKPV	Point-Clef Préparatoire à la Validation
PPF	Product Processing Facility
PPS	Product Processing Software
SIF	Simulateurs d'Interfaces du F-PAC ENVISAT (sous système du CNES)
TES	Time Event Server
TU	Test Unitaire
WO	Working Order
WOM	Work Order Manager

### Liste des paramètres AC :

page D277 §D.

**Liste des paramètres AD :**

page 17 § 3.2.3.

page 26 § 3.5.2.1.

page 28 § 3.5.2.1.

page 193 § 5.2.49.

## 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

Le présent document constitue le plan d'essai du logiciel OPS-IASI.

Ce document sera enrichi pendant toute la durée du projet. Il est livré pour :

RCP	en version préliminaire
la Recette V0 – 1 mois	en version définitive pour recetter la version 0
au PKPV	en version définitive

Il est utilisé pour les tests d'**intégration** et de **validation** effectués par THALES IS.

Ce document a pour but de définir une stratégie d'essais et les procédures de tests nécessaires pour démontrer que le logiciel OPS-IASI est conforme aux spécifications techniques de besoin.

Le Plan d'Essai comporte les chapitres suivants :

- le chapitre 3 présente la phase de tests selon les axes suivants :
  - organisation et planification,
  - ressources humaines, responsabilités,
  - ressources matérielles et logicielles nécessaires,
- le chapitre 4 décrit les procédures et jeux d'essais nécessaires pour effectuer les tests et la traçabilité des tests vis à vis des exigences;
- l'annexe A propose un formulaire de description de test (but du test, procédure, résultat attendus, etc..)
- l'annexe B propose une fiche journal d'essai qui permet de décrire les résultats obtenus et d'indiquer leur conformité par rapport aux résultats attendus.
- l'annexe C fournit la matrice de couverture entre les interfaces internes définies dans le document de Conception Préliminaire [DA108] et les tests d'intégration.
- l'annexe D fournit la matrice de couverture entre les exigences définies dans le Dossier de Spécifications [DA104] et les tests de validation
- l'annexe E fournit la matrice de couverture entre les tests de validation et les exigences définies dans le Dossier de Spécifications [DA104].

## 2.2.DOMAINE D'APPLICATION

Le présent Plan d'Essais est applicable à l'ensemble du logiciel développé dans le cadre de l'affaire **OPS-IASI Level1**. Cette affaire est développée pour **le CNES**, sous la responsabilité du département Espace deTHALES IS.

Les travaux sont effectués dans le cadre du contrat 779/CNES/2001/8937/00.

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

## 2.3.PRESENTATION GENERALE

### 2.3.1.Contexte

IASI est l'un des instruments embarqués sur la future plate-forme METOP (lancement prévu mi-2005) de EUMETSAT. Son rôle est de fournir des profils verticaux de température d'humidité et de concentration en certains gaz à des fins de météorologie. Son apport réside surtout dans la précision des mesures qui doivent atteindre des niveaux inégalés (1 K sur les profils de température et 10% sur les profils d'humidité).

L'instrument IASI est en fait constitué de 2 instruments:

- un **interféromètre de Michelson** effectuant des mesures de spectres,
- un **imageur infrarouge** permettant de générer les produits IASI level 1 et de localiser les images IASI relativement aux images AVHRR (autre instrument embarqué sur METOP).

Les mesures brutes de l'instrument IASI sont réceptionnées puis traitées par le Segment Sol d'EPS appelé Core Ground Segment (CGS), afin de générer les produits IASI de niveau 1 et 2.

La génération des divers types (GOME, IASI, ATOV, GRAS, ASCAT) et niveaux de produits d'EPS étant inter-dépendante et contrainte temporellement, une supervision de la production des données EPS est nécessaire. Elle est confiée à un sous système spécialisé appelé le Product Generation Facility (PGF), en charge de synchroniser la génération des produits. L'architecture de traitement des données EPS a été conçue sur le principe d'un pipeline de type "data-driven". Dans cette architecture, chaque tâche est indépendante et son déclenchement est conditionné seulement par la disponibilité des données d'entrée de la tâche. Cette approche simplifie la mise en œuvre des mécanismes de parallélisme et de synchronisation des chaînes de production à travers la description d'un data flow.

Chaque chaîne de production est encapsulée dans un sous-système appelé Product Processing Facility (PPF). Le PGF a en charge la gestion de la production globale du CGS à travers la synchronisation des PPF.

La configuration cible du PGF est un SP3 où tous les PPF sont installés sur chacun des nœuds du SP3. Le PGF gère la répartition de la puissance de calcul de chaque nœud entre les PPF à travers l'attribution d'un nombre de threads à chaque instance de PPF. Ainsi, plusieurs instances de PPF différents peuvent s'exécuter en parallèle, se partageant les processeurs du nœud d'exécution.

L'Operational Software (OPS) IASI Level 1, est le PPF du PGF chargé de la production des produits de niveau 1 IASI à partir des données de niveau 0 et des images AVHRR. Il intègre la chaîne de traitement qui à partir des données brutes de l'instrument IASI (niveau 0), effectue l'étalonnage radiométrique, la localisation et la datation des images, le rééchantillonnage et l'apodisation des spectres ainsi qu'une analyse croisée AVHRR/IASI afin d'obtenir les produits de niveau 1A et/ou B et/ou C IASI.

Le schéma ci-dessous situe OPS IASI Level 1 dans le l'architecture d'EPS.

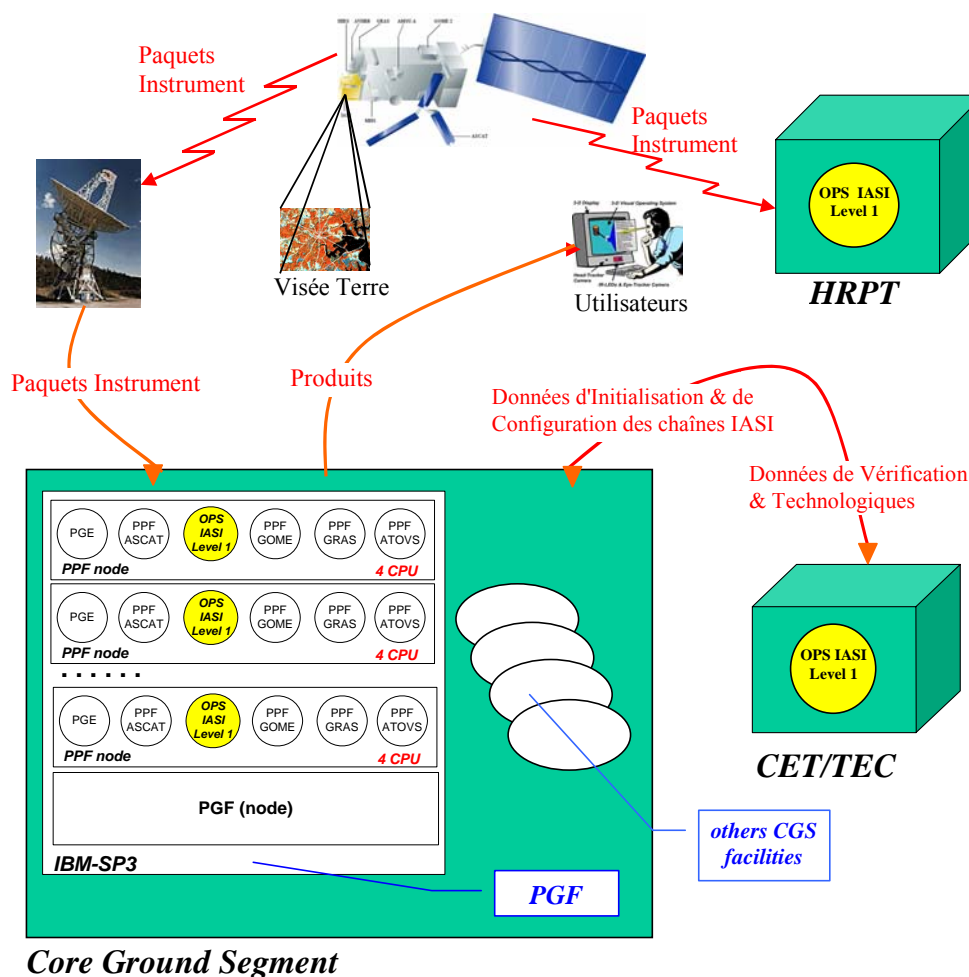


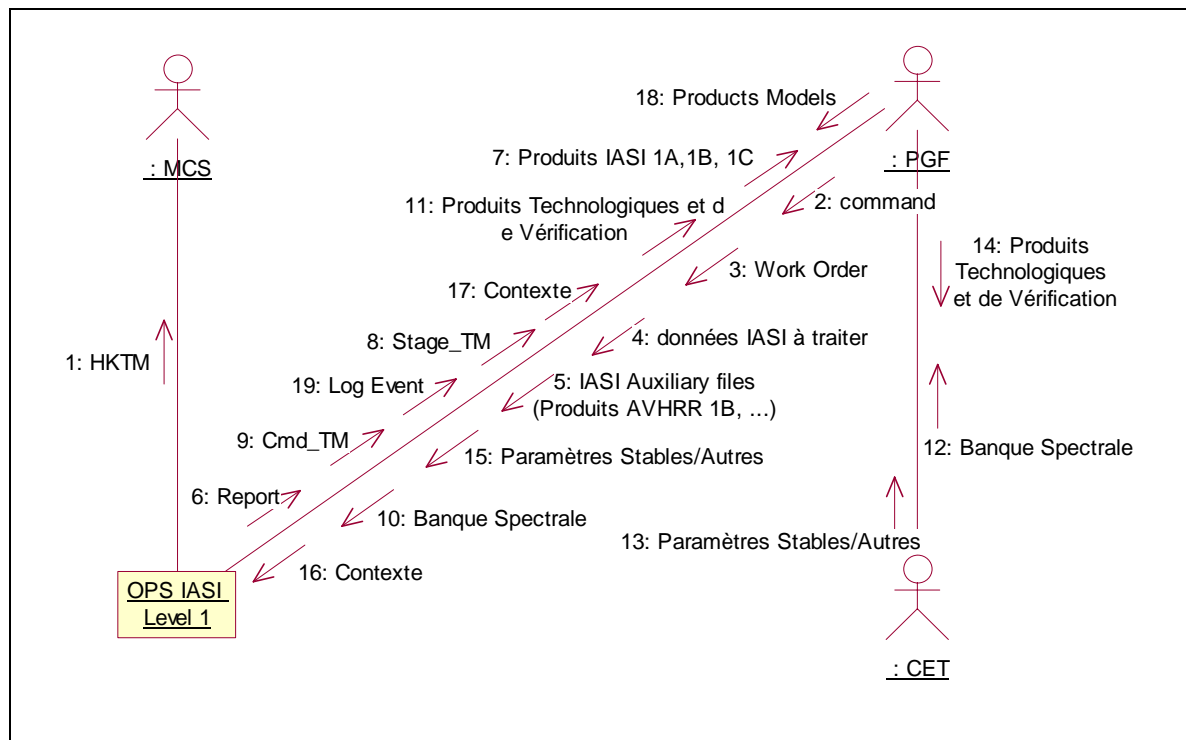
Figure 1 : L'OPS IASI Level1 dans l'architecture EPS

L'OPS IASI Level 1 est livré au CNES afin d'être intégré dans le segment-sol opérationnel CGS. De plus, l'OPS IASI est installée dans le Centre d'Expertise Technique IASI au CNES de Toulouse, ainsi que dans les Centres de Traitement Locaux (HRPT).



### 2.3.2. 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 2 : Diagramme de Contexte de l'OPS-IASI**

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

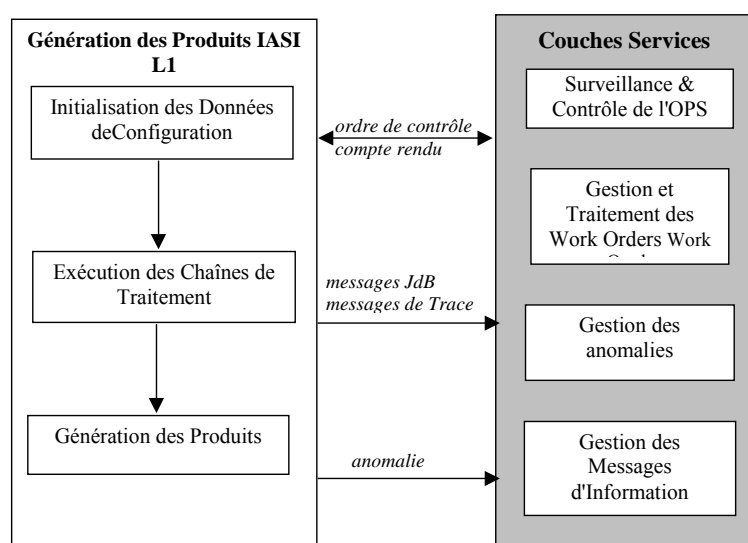
- le MCS qui est le système de Monitoring&Control des sous-systèmes du CGS. Le rôle du MCS est de surveiller le fonctionnement et la disponibilité des divers sous-systèmes du CGS. Pour ceci, les sous-systèmes dont l'OPS doivent lui fournir régulièrement ou sur anomalie des informations de suivi : les 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).

### 2.3.3. Rappel des fonctionnalités

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.



**Figure 3 : Fonctions de l'OPS IASI L1**

## 3. PREVISION DE TESTS

### 3.1. BUT DE CHAQUE PHASE

La phase d'**Intégration** permet de vérifier que les interfaces internes entre les différents composants du logiciel OPS-IASI, sont bien respectées. Le CNES ne participe pas à cette phase.

La phase de **Validation** permet de s'assurer que toutes les exigences mentionnées dans les spécifications Logicielles [DA104] sont prises en compte et correctement implémentées. Elle est sous la responsabilité de THALES IS. Le CNES sera informé de l'état d'avancement des tests (le CNES ne participe pas à cette phase).

### 3.2. STRATEGIE DE TEST

#### 3.2.1. Identification des Interfaces Internes

Le logiciel OPS-IASI est une application multi processus. Les interfaces internes sont identifiées et décrites dans le Dossier de Conception Préliminaire [DA108].

#### 3.2.2. Stratégie d'Intégration

La phase d'intégration consiste à construire par des assemblages successifs d'éléments (classes/modules ou processus) le logiciel final puis à valider le traitement des interfaces inter processus.

Deux niveaux d'intégration sont mis en œuvre dans le cadre de l'OPS :

- l'intégration de niveau processus : dans le cas d'un processus complexe composé d'un grand nombre de classes, une phase d'intégration formelle des classes par groupe fonctionnel pour construire le processus final est nécessaire. Ce type d'intégration est mis en œuvre pour construire chaque étape ou action (au sens tâche à exécuter par un thread du SD) de la chaîne algorithmique. Puis ces actions sont intégrés dans l'architecture multithread afin de constituer le processus SD.
- l'intégration de niveau système : dans le cas de processus constitués à partir de classes réutilisées ou composés de peu de classes, l'assemblage des classes s'effectue durant le test unitaire de niveau processus. En effet, ce type de réalisation est confié à un seul développeur, ce qui minimise les risques d'anomalie d'interfaçage entre les objets d'un processus étant donné que celui-ci a en charge tous les tests depuis les tests unitaires jusqu'au test d'intégration process. L'intégration système consiste alors à assembler les processus afin de construire le logiciel final. Ce type d'intégration est utilisé pour

l'ensemble des processus constituant la structure d'accueil de la chaîne de traitement qui dérivent du SIF : MP, TES, JDBS, MSGS.

L'intégration de niveau processus est mise en œuvre dans le cadre de la réalisation de la V1 afin de construire le processus Serveur de Données final.

L'intégration de niveau système est mis en œuvre dans le cadre de la réalisation de la V0 pour laquelle le processus Serveur de Données intégré est un simulateur. L'intégration système n'est pas rejoué dans le cadre de l'intégration de la version V1.

Aucune phase d'intégration formelle n'est prévue pour la version V2.

Les tests d'intégration sont joués en préambule des tests de validation durant la phase de validation.

### 3.2.2.1.Intégration du Serveur de Données

Le Serveur de données est constitué de deux groupes logiques :

- le groupe Architecture Multithread : constitué d'un ensemble de paquetages/classes réutilisés de SSALTO en charge de la gestion de l'architecture multithread et de la répartition des tâches dans celle-ci,
- le groupe Algorithme : constitué de l'ensemble des classes qui implémentent les chaînes algorithmiques et l'accès aux données associées.

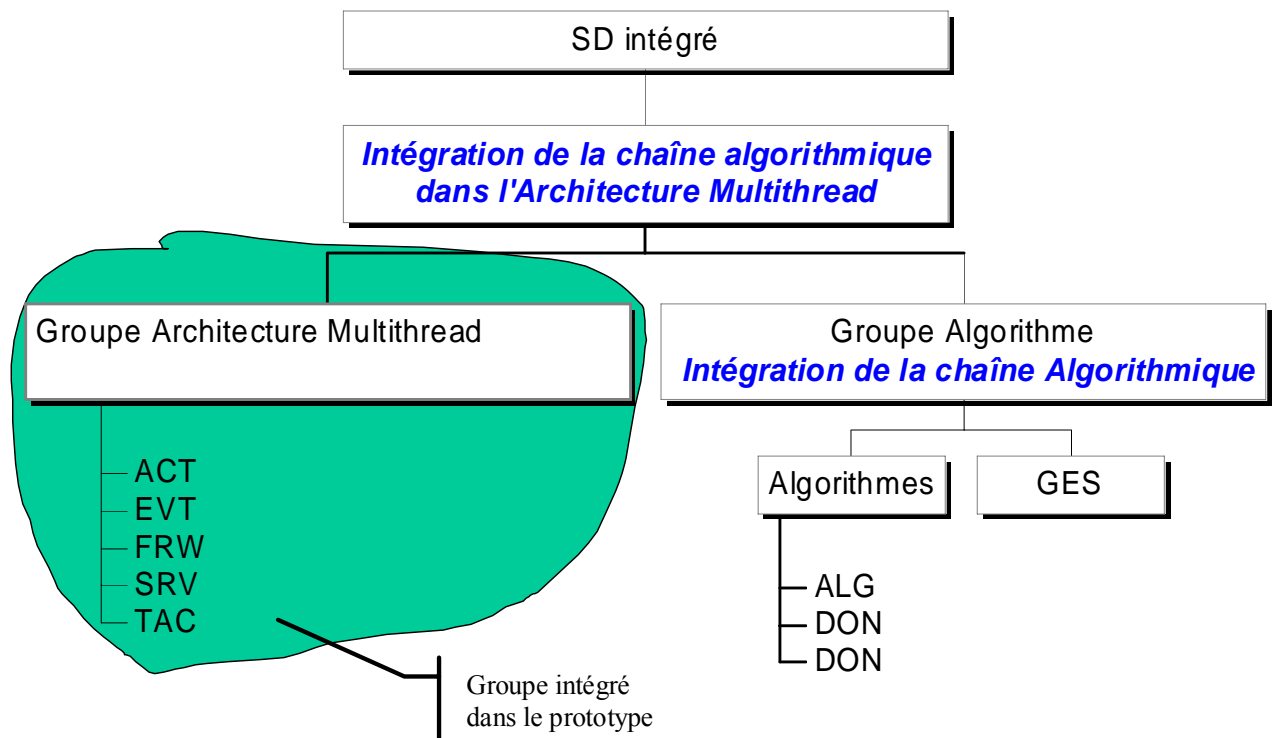
Le premier groupe constitué des paquetages ACT, EVT, FRW, SRV et TAC fait l'objet d'une intégration de niveau système. Ce groupe a été assemblé avec un simulateur d'algorithme afin de constituer le prototype utilisé dans le cadre des essais de performance pendant la phase de spécification. Ce prototype constitue le Serveur de Données utilisé dans la version V0 de l'OPS.

Le second groupe fait l'objet d'une intégration de niveau processus : les sous-paquetages ALG, et DON sont réalisés indépendamment après la réalisation du paquetage BAS. Ils sont intégrés au cours de la phase de codage afin d'obtenir l'ensemble des algorithmes. Cet ensemble est intégré (test d'intégration) avec le sous-paquetage GES afin de constituer les différentes étapes de la chaîne algorithmique; qui elles-même sont intégrées dans l'architecture multithread.

La figure suivante montre la stratégie d'intégration mise en œuvre pour construire le processus Serveur de Données.

Deux tests d'intégration sont définis :

- un test d'intégration de la chaîne Algorithmique : INT-ALGO,
- un test d'intégration de la chaîne algorithmique dans l'architecture multithread : INT-SD.



**Figure 4 : Organigramme d'intégration du Serveur de Données**

### 3.2.2.2. Intégration Système de l'OPS

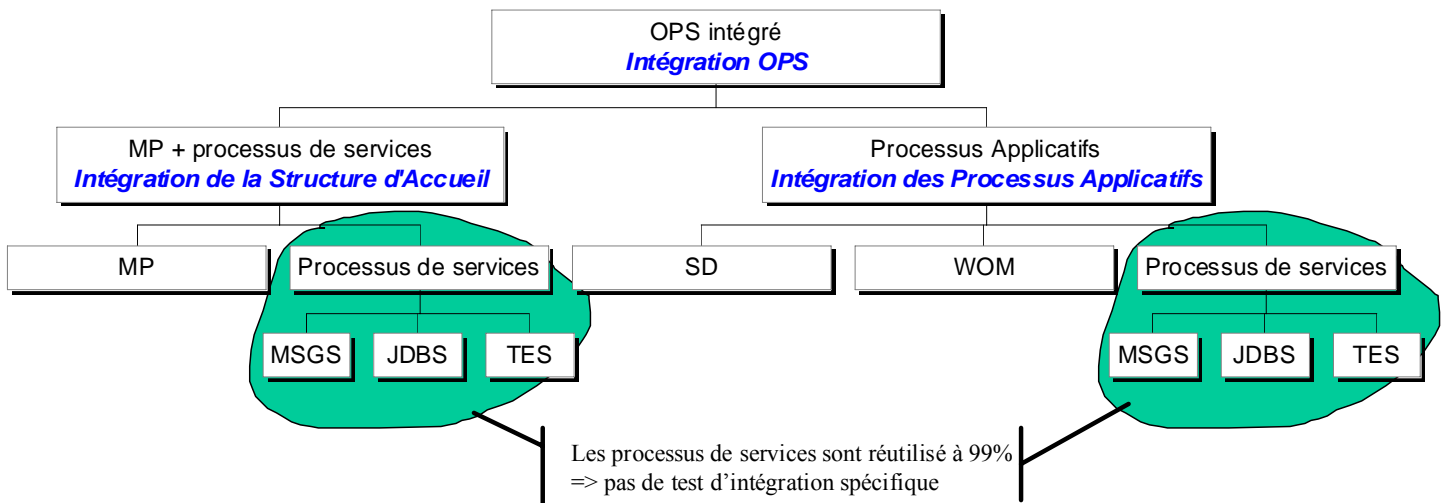
Un ordre de développement des processus est défini afin de faciliter les tests d'intégration. Cet ordre est le suivant : MP, SD et WOM (MSGs, JDBS et TES sont récupérés à 100% du SIF).

Les processus de services (MSGs, TES et JDBS) sont réutilisés à 99%, il ne font donc pas l'objet d'un test d'intégration spécifique; leur intégration est vérifiée au cours de la phase d'intégration.

Trois tests d'intégration sont définis pour assembler les processus qui constituent l'OPS :

- un test d'intégration de la structure d'accueil : INT-STRU,
- un test d'intégration des processus applicatifs : INT-APPL
- un test d'intégration de l'OPS complet : INT-OPS.

La figure suivante présente l'organigramme d'intégration système de l'OPS.



**Figure 5 : Organigramme d'intégration de l'OPS**

### 3.2.3.Stratégie de Validation

Compte tenu des plannings de mise à disposition des sous-systèmes à Eumetsat et du fort contenu algorithmique de l'OPS IASI qui implique une validation longue et minutieuse, le CNES a défini une stratégie de livraison de l'OPS en 3 étapes :

- une version V0 : cette version permet de tester l'intégration de l'OPS dans le CGS. La validation de cette version consiste donc à vérifier que l'OPS est capable de traiter des interfaces externes nominales. Le contenu des interfaces en entrée et en sortie doit seulement être correct du point de vue syntaxique.
- une version V1 : cette version permet de valider complètement le fonctionnement l'OPS IASI. Les résultats algorithmiques obtenus doivent être de bonne qualité.

La validation V1 consiste à vérifier le fonctionnement informatique du sous-système OPS-IASI. Cette vérification a pour objectif de valider l'ensemble des exigences de besoin du CNES et d'EUMETSAT à partir des jeux de données livrés par le CNES ou générés par THALES IS.

Une vérification grossière des produits doit être effectuée afin de ne pas fournir des produits aberrants. Cette vérification consiste à comparer les données essentielles produites par IASI-L1, à savoir la température de brillance spectre et image, aux valeurs de référence correspondantes (fournies par le CNES). L'erreur sur ces données doit être inférieure à 2K sachant que les chaînes algorithmiques n'apportent qu'une très faible correction de la valeur en entrée : une erreur de 2K doit mettre en évidence uniquement des problèmes triviaux de codage des algorithmes et non des problèmes liés à la précision des calculs (pas d'effet de boule de neige).

A la demande du CNES, THALES IS ne joue en validation V1 que les tests sur les jeux de données J1, J1bis, J1quar, J2.9, J5 et J10 et J12ter. Cette restriction, implique que la validation V1 ne couvre que partiellement les exigences de besoin (la liste des tests joués en validation et recette V1 est présentée au §3 du cahier de recette).

- une version V2.0 : cette version permet de valider l'algorithmie de l'OPS en utilisant des seuils sur différentes sorties issues des algorithmes (cf 3.5.5). De plus ici l'ensemble des cas fonctionnels est passé en testant notamment un nombre de cas dégradés important sur les interfaces.

De plus, les tests reportés de la V1 à la V2 sont joués dans le cadre de la validation V2.

- une version V2.1 : cette version permet de valider complètement l'algorithmie de l'OPS. Les tests de validation associés ne portent donc que sur ce point. En V2.1, le logiciel OPS est considéré complètement validé du point de vue fonctionnel (les tests de validation de la V1 ne sont pas ré exécutés).

La validation V2 de l'OPS consiste à valider scientifiquement les produits générés par la version 1 de l'OPS. Il n'existe aucune différence fonctionnelle entre les versions 1 et 2 de l'OPS.

En validation V2, seuls les tests algorithmiques de la validation V1 sur les jeux de données V2-J1 à V2-J9 (cf DA48) sont joués. Un contrôle fin des produits générés est effectué afin de s'assurer de leur qualité numérique. Le seuil d'erreur pour déclarer une valeur de bonne qualité par rapport aux valeurs de référence fournies par le CNES est [AD].

Les tests de validation assurent un taux de couverture de 100% des exigences applicables à la réalisation de l'OPS telles que décrites dans le Dossier de Spécifications Logicielles [DA104].

Les tests exécutés pour la validation de la version V0 appartiennent aux classes de tests suivantes (cf §4.2) :

- BUILD-GEN : génération du produit V0,
- BUILD-INST : installation du produit V0,
- FUNCT-RUN : mise en exécution de l'OPS,
- FUNCT-GRA : simulation de traitement d'un granule,
- FUNCT\_DUMP : simulation de traitement d'un dump complet
- FUNCT-LOG : envoi de messages JdB,
- FUNCT\_ERR : traitement des anomalies,
- M&C-CMD : traitement des commandes du PGF,
- M&C-HKTM : génération de HKTM simulé,
- OPER\_DBG : fonctionnement en mode debug (investigation),
- OPER\_STD : fonctionnement en mode standalone.

### 3.3.ORGANISATION ET PLANIFICATION

Les tests d'intégration et de validation se déroulent dans les locaux de THALES IS.

La phase d'intégration commence dès la fin du codage/TU des process. Un ordre de développement des processus est défini afin d'effectuer des tests d'intégration au plutôt. La phase d'intégration concerne la V0.

La phase de validation commence dès la fin de l'intégration de l'ensemble des processus.

Pour la version V1, le PKPV autorise le démarrage de la phase de validation. Une CRE interne de fin de validation permettra de statuer sur l'état du produit.

### 3.4.RESSOURCES HUMAINES ET RESPONSABILITES

L'intégration puis la validation sont réalisées par l'équipe de développement (THALES IS), sous la conduite du responsable d'intégration / validation (THALES IS).

La préparation du contexte d'essais est sous la responsabilité de THALES IS.

Le CNES n'assiste pas à ces phases mais est tenu au courant de leur avancement.



## 3.5.RESSOURCES MATERIELLES ET LOGICIELLES

### 3.5.1.Ressources matérielles

La configuration matérielle nécessaire pour les phases d'intégration et de validation est la suivante :

- version V0 :
  - 1 station de travail bi-processeurs IBM sous AIX 4.3,
  - la machine PGE framework prêtée par le CNES,
- version V1 :
  - 1 station de travail quadri-processeurs IBM sous AIX5.1,
  - la machine PGE framework prêtée par le CNES,
- version V2 :
  - 1 station de travail quadri-processeurs IBM sous AIX5.1,
  - la machine PGE framework prêtée par le CNES.
- version V3 :
  - 1 station de travail quadri-processeurs IBM sous AIX5.1,
  - la machine PGE framework prêtée par le CNES.
- version V4 :
  - 1 station de travail quadri-processeurs IBM sous AIX6.1,
  - la machine PGE framework prêtée par le CNES.
- version V5 :
  - 1 station de travail quadri-processeurs IBM sous AIX6.1,
  - la machine PGE framework prêtée par le CNES.

### 3.5.2.Jeux de Données

Les fichiers de données et de configuration, utilisés pour les tests d'intégration/validation algorithmiques sont fournis par le CNES. Ces fichiers sont utilisés par THALES IS pour construire les jeux de données complémentaires pour les tests de Monitoring & Control et de robustesse.

Les données d'entrée fournies sont regroupées dans le tableau suivant :

DESCRIPTION DES DONNEES	PROVENANCE
1/ Fichiers de données IASI L0	CNES
2/ Données AVHRR 1B (associé à 1/)	CNES
3/ Banque Spectrale (associé à 1/)	CNES
4/ Les Fichiers de Données Auxiliaires (associé à 1/)	CNES
5/ Paramètres Stables/Autres (associé à 1/)	CNES
6/ 1 Work Order (pour exemple)	THALES IS
7/ 1 Product Model 1A, 1B 1C (pour exemple)	THALES IS

**Tableau 1 : Fourniture : données pour les tests d'Intégration et de Validation**

Les dates de besoin de ces données sont les suivantes :

VERSIONS	DONNEES FOURNIES	DATE AU PLUS TARD
V0	1 jeux de données J0 est fourni par le CNES	Cf. Plan d'Application [DA102]
V1	1 Les jeux de données J1 à J15 spécifiés dans l'annexe 6.1 de DA2 sont fournis par le CNES (sauf indication contraire)	Cf. Plan d'Application [DA102]
V2	Les jeux de données V2-J1 à V2-J9 spécifiés dans le document DA48 sont réutilisés	Cf. Plan d'Application [DA102]

**Tableau 2 : Dates de livraison des Jeux de Données**

On regroupe ici l'ensemble des données utilisées dans le cadre de la validation. Pour chacune, l'origine (CNES ou THALES IS) de la donnée est précisée. De plus, dans certains cas, nous avons indiqué les spécificités des jeux de données ainsi que le numéro du jeu CNES associé (cf document [DA2]). La référence indiquée ici sera utilisée par la suite dans les procédures d'essai.

Les données complémentaires à chaque test (product model, work-order) générées par THALES IS ne sont pas ici précisées.

### 3.5.2.1. Jeux de Données V0 – V1

Nature	Référence données	Référence CNES	Description	Remarques
produit IASI N0	TV_CNE_IA0_0	J0	1 dump nominal de 41 granules de 3' (121 minutes)	Jeu de non régression lié à l'évolution des formats d'interface (conf, N1, AVHRR) . Le premier granule commence en début de ligne et les granules suivants intègrent un overlap suffisant pour assurer la continuité du produit.
produit IASI N0	TV_CNE_IA0_1	J1	1 dump nominal = 1 orbite en mode mesure, découpée en granules de 3'	Collage d'une séquence S0 (30 lignes) de données IASI. Le dernier granule dure 3 lignes IASI. L'Overlap de 8s est intégré, et les granules ne commencent pas exactement en début de ligne IASI
produit IASI N0	TV_CNE_IA0_2	J <sub>1bis</sub>	Idem J <sub>1</sub> , mais réduit à une demi-orbite et sans découpage : le dump est fourni en un seul produit	Le niveau 1 sera fourni (un seul dump)
produit IASI N0	TV_CNE_IA0_3	J1ter	Idem J <sub>1</sub> , réduit à 2 granules, mais avec une autre datation des données	Seules les dates contenues dans le MPHR et les MDR seront modifiées (ajout de 30 secondes à la date initiale)
produit IASI N0	TV_CNE_IA0_4	J <sub>1cinq</sub>	Idem J <sub>1</sub> , réduit à 2 granules, Le fichier de N0 contient des DMDR (Dummy Data Record)	Pas de niveau 1 spécifique à fournir
produit IASI N0	TV_CNE_IA0_5	J <sub>2.1</sub>	Cas dégradé : 1 pixel manquant (i.e. le paquet PX est vide (spectre à 0))	1 granule de 22 lignes (L.) L <sub>1</sub> et L <sub>2</sub> : normales L <sub>3</sub> : SN=4,5 PX <sub>1</sub> absent ; SN=6,7 PX <sub>2</sub> absent ; SN=7,8 PX <sub>3</sub> absent ; SN=9,10 PX <sub>4</sub> absent L <sub>4</sub> , L <sub>5</sub> , L <sub>6</sub> : PX <sub>1</sub> absent à partir de L <sub>7</sub> : normales. Le niveau 1 sera fourni pour comparaison
produit IASI N0	TV_CNE_IA0_6	J <sub>2.1-bis</sub>	Cas dégradé : 1 pixel manquant (i.e. le paquet PX est manquant)	1 granule de 22 lignes (L.) L <sub>1</sub> et L <sub>2</sub> : normales L <sub>3</sub> : SN=4,5 PX <sub>1</sub> absent ; SN=6,7 PX <sub>2</sub> absent ; SN=7,8 PX <sub>3</sub> absent ; SN=9,10 PX <sub>4</sub> absent L <sub>4</sub> , L <sub>5</sub> , L <sub>6</sub> : PX <sub>1</sub> absent à partir de L <sub>7</sub> : normales. Le niveau 1 sera fourni pour comparaison

Nature	Référence données	Référence CNES	Description	Remarques
produit IASI N0	TV_CNE_IA0_7	J <sub>2.2</sub>	Cas dégradé : 2 pixels manquants (i.e. paquets PX manquants)	1 granule de 9 L. IDegPixMiss = 10 sL <sub>1</sub> et L <sub>2</sub> normales L <sub>3</sub> : SN=4,5,6 : PX <sub>1</sub> et PX <sub>2</sub> absents ; SN=7,8,9 : PX <sub>3</sub> et PX <sub>1</sub> absents ; SN=10,11,12 : PX <sub>3</sub> et PX <sub>4</sub> absents L <sub>4</sub> à L <sub>8</sub> : PX <sub>2</sub> et PX <sub>3</sub> absents L <sub>9</sub> normale. <i>Le niveau 1 ne sera pas fourni car ce n'est pas lui que l'on vérifie</i>
produit IASI N0	TV_CNE_IA0_8	J <sub>2.3</sub>	Cas dégradé : trou de données	1 granules de 22 L. IDegSpecGap = 10 sL <sub>1</sub> et L <sub>2</sub> normales L <sub>4</sub> absente L <sub>5</sub> normale L <sub>6</sub> à L <sub>10</sub> absentes L <sub>11</sub> normale. <i>Le niveau 1 ne sera pas fourni car ce n'est pas lui que l'on vérifie</i>
produit IASI N0	TV_CNE_IA0_9	J <sub>2.4</sub>	Cas dégradé : overflow/underflow	1 granules de 22 lignes ; IDegBandMiss=10 sL <sub>1</sub> et L <sub>2</sub> normales L <sub>3</sub> : SN=1 overflow sur 1 échantillon de PX <sub>1</sub> ; SN=2 : underflow sur 2 échantillons de PX <sub>2</sub> L <sub>4</sub> à L <sub>6</sub> : overflow sur un échantillon de P1 L <sub>7</sub> normale. <i>Le niveau 1 ne sera pas fourni car ce n'est pas lui que l'on vérifie</i>
produit IASI N0	TV_CNE_IA0_10	J <sub>2.5</sub>	Cas dégradé : paquet AP	1 granules de 22 lignes ; IDegApMiss=10 sL <sub>1</sub> et L <sub>2</sub> normales L <sub>3</sub> : pas de paquet APL <sub>4</sub> : normales L <sub>5</sub> à L <sub>9</sub> : Pas de paquet APL <sub>10</sub> normale. <i>Le niveau 1 ne sera pas fourni car ce n'est pas lui que l'on vérifie</i>
produit IASI N0	TV_CNE_IA0_11	J <sub>2.6</sub>	Cas dégradé : pas d'image de calibration corps noir IP32 et IP33	1 granules de 22 lignes ; IDegIpbbMiss=10 sL <sub>1</sub> et L <sub>2</sub> normales L <sub>3</sub> : pas de paquets IP <sub>32</sub> ni IP <sub>33</sub> L <sub>4</sub> : normales L <sub>5</sub> à L <sub>9</sub> : Pas de paquet IP <sub>32</sub> ni IP <sub>33</sub> L <sub>10</sub> normale. <i>Le niveau 1 sera fourni pour comparaison</i>

Nature	Référence données	Référence CNES	Description	Remarques
produit IASI N0	TV_CNE_IA0_12	J <sub>2.7</sub>	Cas dégradé : pas d'image de calibration espace froid IP35 et IP36	1 granules de 22 lignes ; IDeglpcsMiss=10 sL <sub>1</sub> et L <sub>2</sub> normalesL <sub>3</sub> : pas de paquets IP <sub>35</sub> ni IP <sub>36</sub> L <sub>4</sub> : normales L <sub>5</sub> à L <sub>9</sub> : Pas de paquet IP <sub>35</sub> ou IP <sub>36</sub> après L <sub>10</sub> normale. <i>Le niveau 1 sera fourni pour comparaison</i>
produit IASI N0	TV_CNE_IA0_13	J <sub>2.8</sub>	Cas dégradé : pas d'image de mesure	1 granules de 22 lignes ; IDeglpmesMiss=10 sL <sub>1</sub> et L <sub>2</sub> normalesL <sub>3</sub> : pas de paquets IP pour SN=5 à 8L <sub>4</sub> : normalesL <sub>5</sub> à L <sub>9</sub> : Pas de paquet IP pour SN=1 à 30L <sub>10</sub> normale. <i>Le niveau 1 ne sera pas fourni car ce n'est pas lui que l'on vérifie</i>
produit IASI N0	TV_CNE_IA0_14	J <sub>2.9</sub>	Cas dégradé : pas de fichier AVHRR	1 dump de 5 granules de 22 lignesIDegRadAvhrrMiss=50 s g1 : normalg2 : pas de fichier AVHRRg3 : normalg4 à g6 : pas de fichier AVHRRg7 : normal. <i>Le niveau 1 ne sera pas fourni car ce n'est pas lui que l'on vérifie</i>
produit IASI N0	TV_CNE_IA0_15	J <sub>2.10</sub>	Cas dégradé : pas d'image AVHRR	1 granules de <del>22</del> 40 lignes ; ldegRadAvhrrMiss= <del>4</del> 40 sL <sub>1</sub> et L <sub>2</sub> normalesL <sub>3</sub> : <del>pas une ligne de</del> <del>données AVHRR pour SN=5 à</del> <del>8</del> <u>manque au milieu de cette ligne</u> L <sub>4</sub> à L <sub>13</sub> : normalesL <sub>14</sub> à L <sub>9</sub> : <del>Pas de paquet</del> <del>IP pour SN=1 à 30L<sub>10</sub> normale de</del> <u>ligne AVHRR pour toute cette ligne</u> <u>(trou de 48 lignes)Après L<sub>15</sub> les lignes</u> <u>sont normales.</u> <i>Le niveau 1 ne sera</i> <i>pas fourni car ce n'est pas lui que l'on</i> <i>vérifie</i>  <i>(on prendraici le premierN0 du</i> <i>premier jeu J2.9)</i>
produit IASI N0	TV_CNE_IA0_16	J <sub>2.11</sub>	Cas dégradé : pas de localisation AVHRR	1 granules de <del>4</del> 22 lignes L <sub>1</sub> et L <sub>2</sub> normalesL <sub>3</sub> : pas de données AVHRR pour SN=5 à 8 <u>Après L<sub>4</sub> les lignes sont</u> normale. <i>Le niveau 1 ne sera pas</i> <i>fourni car ce n'est pas lui que l'on</i> <i>vérifie.</i> <u>Pas de localisation AVHRR ó</u> <u>le flag XXXX est levé</u>  <i>(on prendraici le premierN0 du</i> <i>premier jeu J2.9)</i>

Nature	Référence données	Référence CNES	Description	Remarques
produit IASI N0	TV_CNE_IA0_17	J <sub>2.12</sub>	Cas dégradé : manque de paquet VP	1 granule de 22 lignes L1 : normale L2 : pas de VPAL L3 : pas de VPBL L4 : pas de VPCL L5 : pas de VPD L6 : pas de VPE L7 Pas de VPA, VPB, VPC, VPD, VPEL L8 normale. Le niveau 1 ne sera pas fourni car ce n'est pas lui que l'on vérifie
produit IASI N0	TV_CNE_IA0_18	J <sub>4quar</sub>	1 jeu sur 4g.	J <sub>4bis</sub> = J <sub>4</sub> avec un overlap trop faible pour que la première ligne du deuxième granule puisse être incluse dans le niveau 0 fourni Le niveau 1 est celui produit par J <sub>4</sub> Solution choisie : overlap réduit à 0 secondes et début de granule non synchronisé
produit IASI N0	TV_CNE_IA0_19	J <sub>5</sub>	1 jeu sur 7 jours	J <sub>5</sub> = 2 orbites nominales consécutives Il sera possible de faire le test sur une plus longue durée en relançant plusieurs fois l'OPS sur la dernière orbite. La durée de 7 jours n'est pas nécessaire et sera à discuter avec Thales. Le niveau 1 sera fourni pour comparaison
produit IASI N0	TV_CNE_IA0_20	J <sub>6</sub>		granule de N0 issu de J4 en modifiant le PTSI (2 granule) On continue d'appeler J4 = les 4 premiers granules de J1
produit IASI N0	TV_CNE_IA0_21	J <sub>7</sub>	1 jeu image codée sur 8 bits	J <sub>7</sub> = séquence S <sub>1</sub> (une ligne IASI) dupliquée (10 lignes avec IIS sur 8 bits). Le niveau 1 sera fourni pour comparaison
produit IASI N0	TV_CNE_IA0_22	J <sub>7bis</sub>	1 jeu image codée sur 10 bits	J <sub>7bis</sub> = séquence S <sub>1</sub> dupliquée (10 lignes avec IIS sur 10 bits). Le niveau 1 sera fourni pour comparaison
produit IASI N0	TV_CNE_IA0_23	J <sub>7ter</sub>	1 jeu image codée sur 12 bits	J <sub>7ter</sub> = séquence S <sub>1</sub> dupliquée (10 lignes avec IIS sur 12 bits) Le niveau 1 sera fourni pour comparaison
produit IASI N0	TV_CNE_IA0_24	J <sub>8</sub>	1 jeu de produit 1A	2 g. issus du traitement de J <sub>1</sub> par l'OPS : S <sub>6</sub> . Le niveau 1 est celui fourni avec J <sub>1</sub>  (à générer par THALES)

Nature	Référence données	Référence CNES	Description	Remarques
produit IASI N0	TV_CNE_IA0_25	J <sub>8bis</sub>	1 jeu de produit 1B	2 g. issus du traitement de J <sub>1</sub> par l'OPS : S <sub>7</sub> . <i>Le niveau 1 est celui fourni avec J<sub>1</sub></i>  (à générer par THALES)
produit IASI N0	TV_CNE_IA0_26	J <sub>9</sub>	1 jeu sur 4 g. : mode plateforme	J <sub>8</sub> = J <sub>4</sub> avec un basculement en mode manœuvre dans le plan en milieu du deuxième granule. <i>Le niveau 1 est celui fourni avec J<sub>1</sub>. Ceci est fait par l'intermédiaire d'un fichier OSV</i>
produit IASI N0	TV_CNE_IA0_27	J <sub>10</sub>	1 jeu sur 8 g. comportant des données de calibration	g0 = 1 granule normal  g1 = 50% normal + 50% calibration corps noir  g2 = 50% calibration 1 <sup>er</sup> espace froid + 50% calibration miroir  g3 = 50% calibration visée Terre + 50% normal  g4 = 50% auxiliary mode + 50% normal  g5 = 50% 2 <sup>ème</sup> espace froid + 50% nadir  g6 = 50% mirror parked + 50% normal  g7 = 50% mirror moving + 50% normal Début des g. synchronisés sur ligne IASI, de même que fin des g. synchronisée sur ligne IASI. <i>Le niveau 1 sera fourni pour comparaison</i>

Nature	Référence données	Référence CNES	Description	Remarques
produit IASI N0	TV_CNE_IA0_28	J <sub>13</sub>	1 dump de données synthétiques permettant de lever des flags dans l'OPS et permettant de valider ponctuellement certains algorithmes. Notamment : utilisation du contexte pour initialiser l'historique des positions de l'axe interférométrique, les spectres extrêmes, les interpolations de paramètres pour les cas dégradés, la table des pixels morts. Utilisation d'une banque spectrale particulière (par exemple avec des 0 et des 1) pour tester : La chaîne produit 1A sans ISRFEM, la chaîne ISRFEM	· 23_SSD : GSsdFlagSpectralShiftNonQual (module de l'harmonique)· 21_SSS : GSssFlagNonSelPix (pas de pixel sélectionné)· 24_IAX : GlaxFlagAxeNonQual (échec fit) division par min(A,B)· 25_FAX : to = 0.0 (division par durée d'oubli) : GFaxAxeNonQual (distance axe RMS > seuil)· 43_ISF : GlsfFlagPdsNonValid· 20_DOC : GDocFlagUnderOverflow : GDocFlagErrorNbBit· 111_MCX : MMcxFlagImagRMSNonValid : MMcxFlagBiasCalRad : MMcxFlagNoiseCalRad· 32_HEC : division par B1· 22_SOS : validation de l'interpolation spline qui n'est pas utilisée à présent· 41_CCS : <b>AD</b> . Le jeu de base est J1, avec entre autre une modification de la configuration
produit IASI N0	TV_CNE_IA0_29	J <sub>14</sub>	Données issues du modèle PFM instrument en vide thermique.	différé ( les tests associés ne sont pas joués en V1)
produit AVHRR 1b	TV_CNE_AVH_0	J0	ensemble de produit AVHRR en coherence avec les granules N0 du jeu J0	44 granules
produit AVHRR 1b	TV_CNE_AVH_1	J1	ensemble de produit AVHRR en coherence avec les granules N0 du jeu J1	nombre à preciser
produit AVHRR 1b	TV_CNE_AVH_2	J2.9	ensemble de produit AVHRR en coherence avec les granules N0 du jeu J2.9	
produit AVHRR 1b	TV_CNE_AVH_3	J2.10	ensemble de produit AVHRR en coherence avec les granules N0 du jeu J2.10	
produit AVHRR 1b	TV_CNE_AVH_4	J2.11	ensemble de produit AVHRR en coherence avec les granules N0 du jeu J2.11	



Nature	Référence données	Référence CNES	Description	Remarques
produit IASI 1a	TV_CNE_IA1_1	J <sub>8</sub>	1 jeu de produit 1A	2 g. issus du traitement de J <sub>1</sub> par le MNP : S <sub>6</sub> (premier et deuxième granule). Le niveau 1 est celui fourni avec J <sub>1</sub>
produit IASI 1b	TV_CNE_IA1_2	J <sub>8bis</sub>	1 jeu de produit 1B	2 g. issus du traitement de J <sub>1</sub> par le MNP (premier et deuxième granule): S <sub>7</sub> . Le niveau 1 est celui fourni avec J <sub>1</sub>
produit IASI 1a/1b/1c/ENG/VER	fichiers références fournis pour comparaison			
fichiers de configuration	TV_CNE_CON_0	J0	conf bord,sol,BS en coherence avec les granules N0 du jeu J0	
fichiers de configuration	TV_CNE_CON_1	J1	conf bord,sol,BS en coherence avec les granules N0 du jeu J1	
fichiers de configuration	TV_CNE_CON_2	J6	conf bord,sol,BS en coherence avec le deuxieme granules N0 du jeu J6 ( on simule ici le changement de PTSI associé au changement de banque spectrale)	a generer à partir de TV_CNE_CON_1 (conf autre) en modifiant le PTSI dans le fichier de conf autre
fichiers de configuration	TV_CNE_CON_4	J11bis	conf bord,sol,BS en coherence avec le deuxieme granules N0 du jeu J6 (changement de banque spectrale celle ci etant absente)	J <sub>11bis</sub> = J <sub>4</sub> avec une référence de banque spectrale différente de celle présente. Un fichier de configuration spécifique sera livré
fichiers de configuration	TV_CNE_CON_6	J <sub>11quar</sub>	1 jeu sur 4g. : pas de fichier de configuration 'paramètres stables'	J <sub>11quar</sub> = J <sub>4</sub> sans fichier de configuration 'paramètres stables'. Pas de niveau requis
fichiers de configuration	TV_CNE_CON_7	J <sub>11cinq</sub>	1 jeu sur 4g. : pas de fichier de configuration 'autres paramètres' correspondant au PTSI	J <sub>11cinq</sub> = J <sub>4</sub> sans fichier de configuration 'autres paramètres' correspondant au PTSI
fichiers de configuration	TV_CNE_CON_8	J <sub>13</sub>	en coherence avec le jeu J13	issu du jeu J1

Nature	Référence données	Référence CNES	Description	Remarques
données auxiliaires	TV_CNE_AUX_1	J0	fichiers OSV,SVM et OBT UTC en coherence avec les granules N0 du jeu J0	Le fichier OBTUTC a changé du fait de la synchronisation avec AMSU
données auxiliaires	TV_CNE_AUX_1	J1	fichiers OSV,SVM et OBT UTC en coherence avec les granules N0 du jeu J1 et J4	
données auxiliaires  (fichier OSV)	TV_CNE_AUX_2	J <sub>9</sub>	1 jeu sur 4 g. : mode plateforme	J <sub>8</sub> = J <sub>4</sub> avec un basculement en mode manœuvre dans le plan en milieu du deuxième granule. <i>Le niveau 1 est celui fourni avec J<sub>1</sub> ( fait par THALES)</i>

Tableau 3 : Jeux de données élaborés par le CNES

Nature	Référence données	Référence CNES pour les autres données	Description	Méthode d'élaboration
produit IASI N0	TV_THA_IA0_1	J1quar	1 premier granule dégradé afin d'etre illisible	troncature du granule afin de rendre impossible la lecture
produit IASI N0	TV_THA_IA0_2	J <sub>15</sub>	Jeu de données non représentatives permettant de tester les levées de flag dans les algorithmes. Les fichiers de configuration associés seront établis en fonction (définition de certaines valeurs seuil AD)	Il s'agit d'un granule de 10 lignes de données simulées complètement irréalistes (issue de J1). Ces données seront générées par l'industriel chargé du développement de l'OPS. <i>Jeu fourni par l'industriel chargé du développement de l'OPS ( à priori rempli de 0 + leve de flag instruments)</i>
produit IASI N0	TV_THA_IA0_3	J1	1 premier granule ou on a modifié la version du format ( > 6.3) afin d'etre illisible	modification du granule afin de rendre impossible la lecture
fichiers de configuration	TV_THA_CON_1	J2.1	fichier de configuration contenant des durée permettant le déclenchement des cas dégradés permanents (IdefDurxxx= durée 2 LN)	par modification du jeu J1( conf autre)

Nature	Référence données	Référence CNES pour les autres données	Description	Méthode d'élaboration
<b>fichiers de configuration</b>	TV_THA_CON_2	J11	fichier de configuration autre contenant une version de format incompatible avec la version de la couche d'accès	par modification du jeu J1( conf autre) ( ou généré par le CNES)
<b>fichier de contexte</b>	TV_THA_CTX_1	J1	fichier de contexte à définir avec le CNES	à générer à partir du document d'interface ce fichier référence le PTSI du fichier configuration autre TV_CNE_CON_1
<b>fichier de contexte</b>	TV_THA_CTX_2	J11ter	fichier de contexte avec une version supérieure à la couche (2.0 par ex)	à générer à partir du document d'interface le contenu est idem à celui de TV_THA_CTX_1
<b>fichier de contexte</b>	TV_THA_CTX_3	J3	fichier de contexte généré après le deuxième granule du jeu J1	à générer à partir du Jeu J1 en validation
<b>fichier de contexte</b>	TV_THA_CTX_4	J8	fichier de contexte généré après le premier granule du jeu J1 ( la reprise est ainsi possible)	à générer à partir du Jeu J1 en validation
<b>fichier de contexte</b>	TV_THA_CTX_5		fichier de contexte à froid	à générer à partir du document d'interface, ce fichier est livré dans les données de conf de l'OPS ce fichier référence le PTSI du fichier configuration autre TV_CNE_CON_1

Tableau 4 : Jeux de données élaborés par THALES IS

### 3.5.2.2. Jeux de Données V2

Les jeux de données utilisés dans le cadre de la validation scientifique sont décrits dans le document ref : IA-TN-0000-2681-NOV Ed. 1 Rev. 0. Leur contenu est ainsi spécifié en fonction des objectifs scientifiques de chacun.. Pour certains jeu, une livraison après modification des formats de produits justifie une référence V3.

Le tableau ci –dessous reprend pour chaque jeu, les types de données livrés ainsi que leurs principales

caractéristiques.

Référence CNES	Contenu	Description	Remarques
J0V3	Produits N0 , Produits N1 de référence, Fichiers de configuration, Produits AVHRR	Jeu de test de référence portant sur une orbite nominale (121 minutes)	Les conditions de traitement portent sur des cas non perturbés (spectres , image)
J0dumpV3	Produit N0 , Produit N1 de référence,	Jeu de test en mode dump de référence portant sur une orbite nominale (121 minutes)	Les conditions de traitement portent sur des cas non perturbés (spectres , image)
J1 V3	Produits N0 , Produits N1 de référence, Fichiers de configuration	Jeu de test portant sur la calibration radiométrique des images en prenant en compte des défauts de l'imageur IIS	Changement de PTSI en cours de test et décalages IIS/AVHRR. Test de l'algorithme DPT
J2 V2	Produits N0 , Produits N1 de référence, Fichiers de configuration,	Jeu de test portant sur la validation de l'algorithme 41_CCS	Cas dégradé sans AVHRR
J3 V2	Produits N0 , Produits N1 de référence, Fichiers de configuration,	Jeu de test portant sur la validation de l'algorithme 41_CCS	Cas dégradé avec 2 canaux AVHRR
J4 V2	Produits N0 , Produits N1 de référence, Produits AVHRR	Jeu de test portant sur la validation de l'algorithme 41_CCS.	Transition jour/nuit
J5 V3	Produits N0 , Produits N1 de référence, Fichiers de configuration,	Jeu de test portant sur la validation de la calibration spectrale et sur la post-calibration radiométrique	

Référence CNES	Contenu	Description	Remarques
J6 V2	Produits N0 , Produits N1 de référence, Fichiers de configuration,	Jeu de test portant sur la validation de la calibration spectrale et sur la post-calibration radiométrique	Test de l'algorithme MCX
J7 V3	Produits N0 , Produits N1 de référence, Fichiers de configuration,	Jeu de test de calibration spectrale portant sur une orbite nominale (121 minutes)	Forte variabilité dans le temps de l'axe interférométrique et de la température corps noir
J9 V3	Produits N0 , Produits N1 de référence, Fichiers de configuration,	Jeu de test portant sur une orbite complète (121 minutes) en mettant en œuvre des phases de calibration externe	Bruit ajouté sur la température de corps noir.
J10 V2	Produits N0 , Produits N1 de référence, Fichiers de configuration,	Jeu de test testant une mode de filtrage basique (température, axe interférométrique)	
J11 V2	Produits N0 , Produits N1 de référence, Fichiers de configuration,	Jeu de test testant un fonctionnement avec 2 sondes	Similaire à J10

### 3.5.3.Ressources Logicielles

Les ressources logicielles nécessaires pour passer les tests d'intégration et de validation sont constituées des éléments listés dans le tableau suivant.

RESSOURCES LOGICIELLES	PROVENANCE	DATE AU PLUS TARD
V0 : PGE bibliothèque XML	CNES THALES IS	Cf. Plan d'Application [DA102]
V1: PGE et ESSL	CNES	Cf. Plan d'Application [DA102]

**Tableau 5 : Ressources logicielles nécessaires aux tests d'Intégration et de Validation****3.5.4. Environnement de la Validation**

Les tests de validation sont définis avec la nomenclature précisée au § 3 ( ex RUN-OPS-N-01).

L'arborescence d'installation de l'OPS fournit le répertoire « data » sous lequel l'environnement de validation est créé lors de l'installation de celui-ci sur la configuration cible.

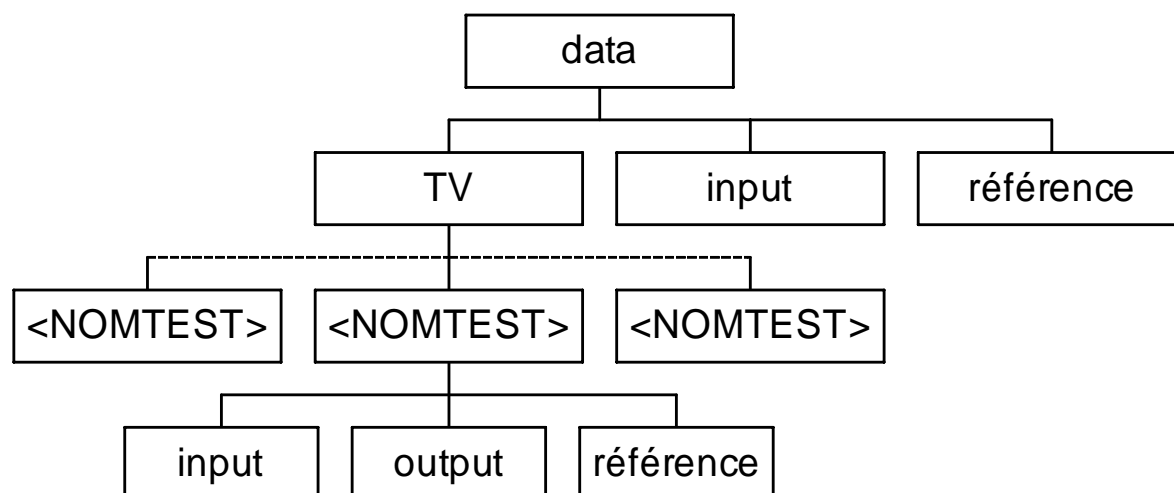
Celui-ci est constitué de la façon suivante :

- un répertoire « input » sous lequel les données d'entrée (N0, AVHRR, fichiers de Configuration et fichiers auxiliaires) sont descendues. Typiquement, c'est sous ce répertoire que seront descendues les données J1 à J15 fournies par le CNES.
- un répertoire « reference » sous lequel les données de référence sont descendues. Typiquement, c'est sous ce répertoire que seront descendues les données de référence des jeux J1 à J15 fournis par le CNES
- un répertoire « TV » sous lequel est stocké pour chaque test de validation (dans un répertoire dédié) l'ensemble de ses données spécifiques (work order, product model, ....) :
  - un répertoire input contenant les entrées spécifiques du test (hors données stockées sous « data/input »),
  - un répertoire output qui permet de stocker temporairement les résultats de chacun des tests en cours (produits ...)
  - un répertoire référence contenant les données de références (issues d'une exécution précédente).

Quand cela est utile, un moniteur (\*.ksh) sera disponible, qui permet d'exécuter automatiquement les différentes étapes du test de validation. Ce moniteur encapsule la mise à disposition des données le lancement du traitement, la récupération des résultats et éventuellement leur comparaison par rapport aux références (remarque le lancement de l'OPS doit être réalisé à priori).

Remarque : dans le cas de données volumineuses (produits) les références sont stockées de manière centralisée et on utilise ici une notion de liens.

La figure suivante, montre l'arborescence de l'environnement de validation.



Les données issues du CNES sont stockées selon leur format de livraison dans un répertoire tmp, 2 niveaux au dessus du working root directory d'exécution. Leur adressage se fait par des liens relatifs à partir du repertoire data/. Celles ci doivent etre installées et décompressées (éventuellement) au préalable à la réalisation des tests de validation. Le contenu des données livrées par le CNES utilisées pour la validation V1 et présentes dans le repertoire tmp est le suivant :

- diskdur\_12mai03
- nov-3112-dt-1450
- nov-3112-dt-1484
- nov-3112-dt-1485
- nov-3112-dt-1491
- nov-3112-dt-1505
- nov-3112-dt-1545

Le contenu des données livrées par le CNES utilisées pour la validation V2 et présentes dans le repertoire tmp est le suivant :

- diskdur\_06aout03
- diskdur\_10juin03
- diskdur\_12mai03
- diskdur\_4juin03

- nov-3112-dt-1484
- nov-3112-dt-1485
- nov-3112-dt-1491
- nov-3112-dt-1505

Les données générées par THALES IS (sorties) sont stockées directement dans le repertoire data du repertoire d'installation .



### 3.5.5.Outils de validation

Les outils de validation mis en œuvre sont de deux types :

- les outils de constitution de jeux de données : ils sont utilisés pour générer à partir des données livrées par le CNES, tous les jeux de validation,
- les outils de vérification de données : ils correspondent aux besoins exprimés dans le document [DA8].
- Les outils permettant de simplifier l'exécution des tests de validation.

L'ensemble de ces outils a été développé par THALES IS mais seuls les outils de vérification et d'exécution des tests de validation sont livrés au CNES. Une aide sommaire est fournie aux utilisateurs en tapant dans un shell « nomoutil » -h. Ces outils sont inclus dans le path de l'utilisateur après exécution du fichier env\_test.sh présent dans le répertoire tools/OPStools

#### 3.5.5.1.Outils de Génération de Données

Outil de génération de contexte de données (TraceContexte) :

**Entrées** : un fichier contexte en ASCII

**Sorties** : un fichier de contexte binaire

**Fonction** : l'outil génère un fichier binaire de contexte à partir de données stockées dans un fichier ASCII.

Outil de modification de fichier de conf (autre) (non livré):

**Entrées** : un fichier de conf autre + un fichier de traces avec position et contenu des variables à modifier

**Sorties** : un fichier de conf autre

**Fonction** : cet outil permet de modifier des données d'un fichier conf autre

Outil de modification de fichier de N0 (non livré):

**Entrées** : un granule de N0 + un fichier de trace avec nom et contenu des variables à modifier ( 1 valeur par variable)

**Sorties** : un granule de N0

**Fonction** : cet outil permet de générer un fichier N0 contenant des données irréalistes issue d'un fichier de trace (toutes les informations générales (datation...) sont conservées) . Cet outil doit permettre d'initialiser les données de base à 0 et de modifier certains bits de la telemesure pour permettre de lever les cas dégradé ( cf annexe 5.3 de la STB).

### 3.5.5.2.Outils de Vérification de Données

#### Outil de dump de produits (TraceProduit):

**Entrées** : 1 produit IASI (N1ABC ENG, VER)

**Sorties** : un fichier au format trace

**Fonction** : cet outil doit pour chaque type de produit (N1A, N1B, N1C, ENG, VER) dumper un record du produit (MPHR, SPHR, GIADR, MDR, VIADR ...). Dans le cas des MDR et VIADR, l'utilisateur peut spécifier le numéro du record visé et l'outil doit sauter les premiers enregistrements. Dans le cas du MPHR un format mot-cle/valeur peut être envisagé. Dans le cas des tableaux, on se ramène systématiquement à une variable à une dimension écrite dans l'ordre du stockage( hormis pour les images 2D).

Dans les Cas des N1 verif il faut pouvoir extraire des MRD complets (hors GRH) (VPx AP) et les dumper dans des fichiers spécifiques

Remarque : les mot clés dans le fichier de trace sont indicés par le numéro du MDR (ligne)

#### Outil de dump du fichier de contexte (TraceContexte):

**Sorties** : un fichier de contexte

**Sorties** : un fichier au format TRACE

**Fonction** : cet outil dumpé dans un fichier ASCII un fichier contexte.

#### Outil de comparaison de fichiers de traces (compfic):

**Entrées** : 2 fichiers au format TRACE

**Sorties** : un fichier au format TRACE

**Fonction** : cet outil permet de comparer chaque variable du fichier avec une variable du même nom dans le fichier de référence. La comparaison se fait en utilisant un critère de précision relatif modifiable. Le nombre total d'anomalies est alors comptabilisé et chaque différence est notée dans le fichier de sortie. Une modification de la précision est possible dans le fichier en entrée en introduisant un mot clé spécifique. .

#### Outil d'analyse de résultats (diff.sh)

**Entrées** : le type du granule à analyser (1A/1B/1C/ENG ) un numéro de granule et un type de comparaison (DETAIL/GLOBAL)

**Sorties** : un fichier de comparaison de nom « output/diff\_type\_numero.txt »

**Fonction** : Cet outil encapsule les outils précédents de la manière suivante . Pour le granule et le type de produit demandé, le produit est dumpé (répertoire output) puis comparé avec le produit de référence (répertoire références) précédemment dumpé (outil dumpref.ksh) en utilisant éventuellement un fichier de précision « input/précision.tra ». Le résultat est analysé de 2

manières :

- Option DETAIL : pour une série de mots clés stockés dans le fichier « input/keylist\_type.txt », les différences sont analysées mot-clé par mot-clé et répertoriées.
- Option GLOBAL : le nombre de différences global et MDR par MDR et VIADR par VIADR est calculés.

### Outils de visualisation de résultats :

**Entrées** : 1 fichiers au format TRACE

**Sorties** : visualisation

**Fonction** : cet outil permet de visualiser sous forme de courbes des données issues d'un fichier de trace. Il permet également de réaliser la synthèse sur la calibration spectrale et radiométrique (rééchantillonnage) précisée dans le document [DA8].

**Utilisation** : taper idl puis @cpfview.sh. ( remarque la variable IDL\_PATH doit etre positionnée (cf env\_test.sh)

- l'outil de visualisation de courbe se lance par la commande « xcurve ».
- l'outil de visualisation d'images se lance par la commande « ximage ».

## 4.DESCRPTION DES TESTS

### 4.1.PRESENTATION DES TESTS

Les tests sont organisés en classes de tests selon le découpage suivant :

- essais de production et d'installation du logiciel,
- essais liés aux exigences fonctionnelles, eux-même subdivisés en :
  - essais liés aux exigences de performances,
  - essais liés aux exigences opérationnelles,
- essais algorithmiques.

Chaque classe d'essai comprend des essais nominaux et des essais dégradés. Les essais nominaux correspondent à un mode de fonctionnement nominal de l'OPS et des sous-systèmes en interface avec lui. Le mode dégradé permet de tester le fonctionnement du logiciel lorsque les interfaces externes ne sont pas respectées ou lorsque les données à traiter son dégradées.

### 4.2.FORMALISME DES TESTS

Chaque test fait l'objet de 2 fiches présentées en annexe.

**La fiche de test** fournit les informations suivantes :

- l'identifiant du test (lié à la classe du test),
- le but,
- les conditions initiales requises, et les critères d'entrée du test,
- la description du test,
- la liste des exigences couvertes,
- le résultat attendu, et les conditions de sortie du test.

**La fiche journal d'essai** contient les résultats des cas de test d'un essai. Dans le cas de test négatif, elle précise les opérations effectuées, les conditions de leur exécution et les résultats obtenus, la description du problème rencontré qui peut se limiter à la référence à une Fiche d'Anomalie. Pour les tests positifs, la fiche mentionne simplement OK.

Chaque cas de test identifié est référencé selon la nomenclature suivante :

<CLASSE\_ESSAI>-<SOUS\_CLASSE\_ESSAI>-<TYPE\_TEST>-<NUMERO\_TEST>

<CLASSE\_ESSAI> : 3 à 5 lettres représentant les 6 grandes classes de tests

<SOUS\_CLASSE\_ESSAI> : 4 lettres(maximum) représentant une sous-classe de  
<CLASSE\_ESSAI>

<TYPE\_TEST> : 1 lettre indiquant le type de test (N : nominal ; D : dégradé)

<NUMERO\_TEST> : 2 chiffres indiquant le numéro du test dans la classe

La liste suivante présente l'ensemble des classes/sous-classes d'essai qui ont été identifiées pour valider l'OPS-IASI :

- **BUILD** essais liés à la production et à l'installation du logiciel :
  - GEN production du logiciel,
  - INST installation du logiciel.
- **FUNCT** essais liés aux exigences fonctionnelles :
  - RUN mise en exécution de l'OPS,
  - CONF configuration des chaînes de traitement de l'OPS
  - GRAN traitement de granules en mode Nominal
  - GRAI traitement de granules en mode Intermédiaire
  - DUMP traitement d'un dump complet
  - OVER gestion de l'overlap
  - COLD utilisation d'un contexte à froid
  - LOG gestion des messages JdB
  - ERR traitement des anomalies et arrêt de l'OPS
- **M&C** essais liés aux exigences de monitoring & Control de l'OPS et des traitements par le PGF :
  - CMD traitement des commandes du PGF
  - HKTM émission de la HKTM
- **PERF** essais liés aux exigences de performances et de robustesse
- **ALGO** essais liés aux exigences algorithmiques
- **OPER** essais liés aux exigences opérationnelles :
  - DBG essais liés au fonctionnement en mode Debug,
  - STD essai lié au fonctionnement en standalone
  - PROC essai de traitement sur 2 processeurs
  - MULT essai en multi instances
  - RAMS essais de RAMS

## 4.3.MODE DE COUVERTURE DES EXIGENCES

### 4.3.1.Exigences validées par Inspection

Un certain nombre d'exigences ne peuvent pas être validées par des tests. Suivant le cas, elles sont donc validées :

- soit par inspection de la Conception,
- soit par inspection du Logiciel (code source),
- soit par le contrôle des travaux effectués lors de la phase de développement,
- soit par analyse des résultats obtenus aux tests de validation.

#### 4.3.1.1.Inspection de la Conception

Les exigences suivantes sont validées par inspection de la Conception et ne font pas l'objet de tests de validation :

Issues de IA-IN-2100-9535-CNE Ed1.0 :

[E\_CTX0] [E\_CTX1] [E\_CTX3] [E\_CTX4bis] [E\_CTX5] [E\_IFS1] [E\_IFS4] [E\_IFS4bis] [E\_IFS6] [E\_IFS7] [E\_INT4] [E\_INT5] [E\_INT6] [E\_INT7] [E\_INI1] [E\_INI6] [E\_MEL1] [E\_MEL1bis] [E\_MEL3] [E\_MEL4] [E\_MEL4ter] [E\_MEL4quar] [E\_MEL8] [E\_LIB1] [E\_LIB2] [E\_LIB3] [E\_LIB3bis] [E\_LIB4] [E\_CON5] [E\_ENC1] [E\_ENC2] [E\_OPR1] [E\_OPI1]

[PPS-3.1-050] [PPS-3.1-060] [PPS-3.1-070] [PPS-3.2-050] [PPS-3.2-070] [PPS-3.2-330] [PPS-3.3-030] [PPS-3.3-040] [PPS-3.3-050] [PPS-3.3-060] [PPS-3.3-080] [PPS-3.3-110pt1] [PPS-3.3-130] [PPS-3.5-090] [PPS-3.5-100] [PPS-3.5-110]

[IF-ALGO-4.4-130]

[SYS-0070] [SYS-0220] [SYS-0230] [SYS-0500] [SYS-0680] [SYS-0690] [SYS-0720] [SYS-0760] [SYS-0790]

[PPF-0050] [PPF-0390] [PPF-0410] [PPF-0610pt2] [PPF-0710] [PPF-0750] [PPF-0780] [PPF-0790] [PPF-0800] [PPF-0810] [PPF-0860] [PPF-0890] [PPF-0900] [PPF-0920] [PPF-0940] [PPF-0950]

Proposées par Thales IS :

[E\_CTX6] [E\_OPI2] [E\_IFS5] [E\_INT3]

[PPS-3.1-030]

[IF-ALGO-4.1-110] [IF-ALGO-4.3-100] [IF-ALGO-4.1-140]

[SYS-0030] [SYS-0250] [SYS-0260] [SYS-0270] [SYS-0280] [SYS-0300] [SYS-0310] [SYS-0320] [SYS-0480] [SYS-0700] [SYS-0740] [SYS-0780] [SYS-0800] [SYS-0810] [SYS-0820] [SYS-0830] [SYS-0840] [SYS-0850] [SYS-0860]

[PPF-0230]

### 4.3.1.2. Inspection du Logiciel

Les exigences suivantes sont validées par inspection du code source livré et ne font pas l'objet de tests de validation :

[E\_MEL2] [E\_MEL7] [E\_STR1] [E\_CTX4bis] [E\_OPC0] [E\_OPC0bis] [E\_OPC1] [E\_OPC2] [E\_OPC3] [E\_OPC4]  
[E\_OPC5] [E\_OPC6] [E\_OPC7] [E\_OPC8] [E\_OPC9]

[PPS-3.3-090]

[PPF-0370]

### 4.3.1.3. Exigences de Développement

Les exigences suivantes sont validées par le suivi des travaux de la phase de développement et ne font pas l'objet de tests de validation :

[E\_INT12] [E\_MEL1] [E\_MEL1bis] [E\_DBG2] [E\_TST1] [E\_TST2] [E\_TST3] [E\_TST4] [E\_TST5] [E\_TST6]  
[E\_TST7] [E\_TST8]

[PPS-3.7-050] [PPS-3.7-080]

[PPF-0110] [PPF-0760]

[PPS-3.7-020]

### 4.3.1.4. Analyse des Résultats des Tests de validation

L'exigence suivante est validée par déduction des résultats obtenus aux tests de validation et ne fait pas l'objet de tests de validation :

[PPS-3.6-060] [PPS-3.7-010]

### 4.3.2.Exigences non Validées

Les exigences suivantes ne sont pas validées soit car elles sont non applicables ou parce que l'OPS n'est pas conforme (cf Dossier de Spécification Logicielle [DA104]) :

#### 4.3.2.1.Exigences Non Applicables

##### Issues de IA-IN-2100-9536-CNE Ed1.0 :

[PPF-0090] [PPF-0100] [PPF-0670] [PPS-3.2-130] [PPS-3.2-150] [PPS-3.2-510] [PPS-3.6-010] [PPS-3.6-070]  
 [SYS-0160] [SYS-0170] [SYS-0173] [SYS-0174] [SYS-0190] [PPS-3.3-070] [PPS-3.4-060] [PPS-3.4-070] [PPS-3.4-080] [PPS-3.4-090] [PPS-3.4-100]  
 [SYS-0173bis] [SYS-0510] [SYS-0520] [SYS-0521] [SYS-0522] [SYS-0523] [SYS-0530] [SYS-0531] [SYS-0540] [SYS-0550] [SYS-0560] [SYS-0570] [SYS-0571] [SYS-0580] [SYS-0590] [SYS-0600] [SYS-0610] [SYS-0620] [SYS-0630] [SYS-0640] [SYS-0641] [SYS-0650] [SYS-0660]  
 [PPF-0520] [PPS-3.7-020 pt2]  
 [SYS-0750] [SYS-0900] [SYS-0910] [SYS-0920] [SYS-0930] [SYS-0940] [SYS-0950] [SYS-0960] [SYS-0970] [SYS-0980] [SYS-0990] [SYS-1000] [SYS-1010] [SYS-1020] [SYS-1030] [SYS-1040] [SYS-1050] [SYS-1060] [SYS-1065] [SYS-1070] [SYS-1080] [SYS-1090] [SYS-1100] [SYS-1110] [SYS-1120] [SYS-1130] [SYS-1140] [SYS-1150] [SYS-1160] [SYS-1170] [SYS-1180] [SYS-1190] [SYS-1200] [SYS-1210] [SYS-1220] [SYS-1230] [SYS-1240] [SYS-1250] [SYS-1260] [SYS-1270] [SYS-1280] [SYS-1290] [SYS-1300] [SYS-1310] [SYS-1320] [SYS-1330] [SYS-1335] [SYS-1340] [SYS-1350] [SYS-1360] [SYS-1370] [SYS-1380] [SYS-1390] [SYS-1400] [SYS-1410] [SYS-1420] [SYS-1430] [SYS-1440] [SYS-1445] [SYS-1450] [SYS-1460] [SYS-1470] [SYS-1480] [SYS-1490] [SYS-1500] [SYS-1510] [SYS-1520] [SYS-1530] [SYS-1540] [SYS-1550] [SYS-1560] [SYS-1570] [SYS-1580] [SYS-1590] [SYS-1600] [SYS-1610] [SYS-1620] [SYS-1630] [SYS-1640] [SYS-1650] [SYS-1700] [SYS-1710] [SYS-1720] [SYS-1730]  
 [IF-ALGO-4.1-120] [IF-ALGO-4.2-110] [IF-ALGO-4.2-130] [IF-ALGO-4.2-150] [IF-ALGO-4.2-160] [IF-ALGO-4.2-170] [IF-ALGO-4.3-110] [IF-ALGO-4.4-181] [IF-ALGO-4.4-190] [IF-ALGO-4.4-200]  
 [SYS-0040] [SYS-0050] [SYS-0171] [SYS-0172] [SYS-0240] [SYS-0400] [SYS-0420] [SYS-0430] [SYS-0450] [SYS-0460] [SYS-0470] [SYS-0180] [SYS-0210] [SYS-0360] [SYS-0410] [SYS-0671] [SYS-0710]  
 [PPF-0218] [PPF-0220] [PPF-0250] [PPF-0701] [PPF-0811]  
 [PPS-3.2-370] [PPS-3.2-140] [PPS-3.4-020] [PPS-3.7-030]

##### Issues de IA-IN-2100-9541-CNE Ed1.0 :

[SYS-0770][SYS-0090] [SYS-0100] [SYS-0670] [PPF-0300][PPS-3.5-120]

##### Proposées par Thales IS :

[SYS-0162] [PPF-0970]



### 4.3.2.2. Non Implémentées par le PGE

[PPS-3.2-020] [PPS-3.3-020] [PPF-0780]

[IF-ALGO-4.1-130]. [IF-ALGO-4.4-100]. [IF-ALGO-4.4-110]. [IF-ALGO-4.4-120] [IF-ALGO-4.4-131]. [IF-ALGO-4.4-140].

[IF-ALGO-4.4-160]. [IF-ALGO-4.4-170]. [IF-ALGO-4.4-180]. [IF-ALGO-4.4-181]. [IF-ALGO-4.4-182]. [IF-ALGO-4.4-183]. [IF-ALGO-4.4-190]. [IF-ALGO-4.4-200]

### 4.3.2.3. Non Conformes

[E\_OPI3]

### 4.3.2.4. Non Testables

Les exigences suivantes ne sont pas validées par des tests car elles ne sont pas testables :

[PPS-3.5-020] [PPS-3.5-030] [PPS-3.5-035pt2]

[SYS-0010] [SYS-0020] [SYS-0730]

[PPF-0830]

[PPF-0530 pt2] : Le CNES n'a pas prévu de fournir un fichier N0 contenant plus qu'une orbite de données.

## 4.3.3. Classe d'Essais BUILD

### 4.3.3.1. GEN : Production du Logiciel

L'objectif des essais de la sous-classe BUILD-GEN est de générer l'ensemble des exécutable qui constituent le logiciel OPS-IASI.

### 4.3.3.2. INST : Installation du Logiciel

L'objectif des essais de la sous-classe BUILD-INST est d'installer le logiciel OPS sur la machine d'exécution.

## 4.3.4. Classe d'Essais FUNCT

### 4.3.4.1. RUN : Lancement de l'OPS

L'objectif des essais de la sous-classe FUNCT-RUN est de vérifier les exigences fonctionnelles liées à la mise en exécution de l'OPS.

### 4.3.4.2. CONF : Configuration de l'OPS

L'objectif des essais de la sous-classe FUNCT-CONF est de vérifier les exigences fonctionnelles liées à la configuration de l'OPS :

- utilisation du contexte,
- chargement des fichiers de configuration,
- vérification de la compatibilité des versions,
- cas dégradés sur les fichiers de configuration,
- .....

### 4.3.4.3. GRAN : Traitement de granules en mode Nominal

L'objectif des essais de la sous-classe FUNCT-GRAN est de vérifier les exigences fonctionnelles liées au traitement en mode nominal d'un granule :

- traitement du premier granule d'un dump,
- traitement d'un granule intermédiaire d'un dump,
- traitement du dernier granule d'un dump,
- test des enchainements entre les modes instruments,
- test des enchainements entre les modes plateformes...
- test des cas dégradés associées aux données d'entrée .....

### 4.3.4.4. GRAI : Traitement de granules en mode Intermédiaire

L'objectif des essais de la sous-classe FUNCT-GRAI est de vérifier les exigences fonctionnelles liées au traitement d'un granule en mode intermédiaire :

- traitement 1A-1C,
- traitement 1B-1C.

#### **4.3.4.5.DUMP : Traitement d'un Dump Complet**

L'objectif des essais de la sous-classe FUNCT-DUMP est de vérifier les exigences fonctionnelles liées au traitement d'un dump complet.

#### **4.3.4.6.OVER : Gestion de l'Overlap**

L'objectif des essais de la sous-classe FUNCT-OVER est de vérifier les exigences fonctionnelles liées au traitement de l'overlap.

#### **4.3.4.7.COLD : Traitement nominal avec initialisation à froid**

L'objectif des essais de la sous-classe FUNCT-COLD est de vérifier les exigences fonctionnelles liées au traitement à partir d'un fichier contexte à froid.

#### **4.3.4.8.LOG : Le Journal de Bord**

L'objectif des essais de la sous-classe FUNCT-LOG est de vérifier les exigences fonctionnelles liées messages JdB pour des situations spécifiques.

#### **4.3.4.9.ERR : Arrêt sur cas dégradé**

L'objectif des essais de la sous-classe FUNCT-ERR est de vérifier les exigences fonctionnelles liées à l'arrêt de l'OPS en cas d'anomalie système grave.

### **4.3.5.Classe d'Essais M&C**

#### **4.3.5.1.CMD : Traitement des Commandes Stop/Break/Suspend/Resume/Abort**

L'objectif des essais de la sous-classe M&C-CMD est de vérifier les exigences fonctionnelles liées au traitement des commandes.

#### **4.3.5.2.HKTM : Génération de la HKTM-Status**

L'objectif des essais de la sous-classe M&C-HKTM est de vérifier les exigences fonctionnelles liées à la génération des HKTM status.

### **4.3.6.Classe d'Essais PERF**

L'objectif des essais de la sous-classe PERF est de vérifier les exigences de performances et de robustesse du sous-système OPS.

### 4.3.7. Classe d'Essais ALGO

L'objectif de cette classe d'essais est de vérifier les exigences algorithmiques du sous-système OPS : traitement des modes instruments, ...

### 4.3.8. Classe d'Essais OPER

#### 4.3.8.1. DBG: Fonctionnement en mode DEBUG

L'objectif des essais de la sous-classe OPER-DBG est de vérifier les exigences de fonctionnement du sous-système OPS en mode debug :

#### 4.3.8.2. STD : Fonctionnement en Standalone

L'objectif des essais de la sous-classe OPER-STD est de vérifier les exigences de fonctionnement en standalone du sous-système OPS.

#### 4.3.8.3. PROC: Fonctionnement sur 2 processeurs

L'objectif des essais de la sous-classe OPER-PROC est de vérifier le fonctionnement du sous-système OPS dans une configuration dégradée où le nombre de processeurs alloué n'est pas nominal.

#### 4.3.8.4. MULT-Exécution de plusieurs instances

L'objectif des essais de la sous-classe OPER-MULT est de vérifier les exigences de fonctionnement du sous-système OPS sur deux nœuds différents du SP3. Cette exigence ne peut être vérifiée que dans l'environnement CGS et non dans l'environnement PGE framework.

## 5.PROCEDURES ET JEUX D'ESSAIS

### 5.1.TESTS D'INTEGRATION

#### 5.1.1.INT-ALGO-N-01

Fiche de Test	INT-ALGO-N-01	OPS
<p><b>But du test :</b></p> <p><i>Construire par étapes successives la chaîne algorithmique et tester les interfaces d'échange entre la couche de gestion des données et la couche algorithmique.</i></p> <p><i>De plus, il est vérifié dans un contexte monothread:</i></p> <ul style="list-style-type: none"> <li>• que les données statiques lues en début de traitement sont accessibles par les algorithmes</li> <li>• que les données dynamiques lues en début de granule sont accessibles par les algorithmes</li> <li>• que les données dynamiques créées par les algorithmes sont accessibles et écrites dans les produits en sortie.</li> </ul>		
<p><b>Conditions initiales requises :</b></p> <p>Au préalable à ce test, les paquetages BAS puis DON et ALG on été développés de manière indépendante. Le paquetage GES permet d'effectuer le lien entre les traitements et la gestion des données au sein du SD . Il est quand à lui préalablement testé unitairement afin de vérifier que l'enchaînement des traitements au sein de chaque étape est correct (cohérence fonctionnelle et prise en compte des cas dégradés).</p> <p>On rappelle ci-dessous l'ensemble des plans de production nominaux de la chaîne OPS :</p> <ol style="list-style-type: none"> <li>1. mode 0-1C : Lecture + étape 1 + étape 2 + étape 3 + étape 4 + écriture</li> <li>2. mode 1A-1C : Lecture + étape 3 + étape 5 + écriture</li> <li>3. mode 1B-1C : Lecture + étape 3 + étape 6 + écriture</li> </ol> <p>Des données statiques et dynamiques cohérentes sont disponibles (données de validation V0)</p>		
Description :	Résultat attendu :	Interfaces testées :
<p>1. Lancer le pilote de test intégrant dans un même processus en mode debug:</p> <ul style="list-style-type: none"> <li>• l'initialisation des données statiques</li> <li>• l'initialisation des données dynamiques,</li> <li>• l'étape 1 en mode debug.</li> </ul>	<p>Vérifier l'intégration de la procédure d'initialisation avec l'étape 1 de la chaîne algorithmique.</p> <p>Vérifier par échantillonnage que les données lues sont correctement reçues par les algorithmes de la première étape dans un contexte monothread .</p> <p>NB : on pourra se baser ici sur des traces spécifiques produites par le logiciel</p>	<p>[f_ParametresAutres] [f_ParametreStables] [f_BanqueSpectrale] [f_DonnéesAuxiliaires] [f_Contexte] [f_ProductsModels] [f_ProduitIASI-In] [f_AlgoTrace]</p>

<p>2. Lancer le pilote de test intégrant dans un même processus :</p> <ul style="list-style-type: none"> <li>• l'initialisation des données statiques,</li> <li>• l'initialisation des données dynamiques,</li> <li>• l'étape 1,</li> <li>• l'étape 2,</li> <li>• l'étape 3,</li> <li>• l'étape 4.</li> </ul>	<p>Vérifier l'intégration des étapes de la chaîne algorithmique.</p> <p>Vérifier par échantillonnage que les données générées par la première étape sont accessibles par la deuxième dans un contexte monothread.</p> <p>NB : on pourra se baser ici sur des traces spécifiques produites par le logiciel</p>	<p>Interfaces internes :</p> <ul style="list-style-type: none"> <li>• par ligne,</li> <li>• par thread,</li> <li>• globales,</li> <li>• contexte courant,</li> <li>• cas dégradés.</li> </ul>
<p>3. Lancer le pilote de test intégrant dans un même processus :</p> <ul style="list-style-type: none"> <li>• l'initialisation des données statiques,</li> <li>• l'initialisation des données dynamiques,</li> <li>• l'étape 3,</li> <li>• l'étape 6,</li> <li>• l'écriture des données.</li> </ul>	<p>Vérifier par échantillonnage que les données lues sont correctement reçues par les algorithmes de la première étape dans un contexte monothread .</p> <p>NB : on pourra se baser ici sur des traces spécifiques produites par le logiciel et sur un dump des produits en sortie</p> <p>Vérifier par échantillonnage que les données générées par la première étape sont accessibles par la deuxième dans un contexte monothread.</p> <p>Vérifier par échantillonnage que les données générées par les étapes de calcul sont prises en compte par les fonction d'écriture de donnée dans un contexte monothread.</p>	<p>[f_Contexte] [f_ProductsModels] [f_ProduitIASI-In] [f_AlgoTrace] [f_ProduitsIASI-1C] [f_DonnéesTechno]</p> <p>Interfaces internes :</p> <ul style="list-style-type: none"> <li>• par ligne,</li> <li>• par thread,</li> <li>• globales,</li> <li>• contexte courant,</li> <li>• cas dégradés.</li> </ul>

## 5.1.2.INT-SD-N-01

Fiche de Test	INT-SD-N-01	OPS
<p><b>But du test :</b></p> <p>Tester l'intégration de la chaîne OPS dans l'architecture multithread. On valide dans ce contexte la communication entre la couche de gestion des données mise en œuvre hors taches et la couche algorithmique multithreadée.</p> <p>Il s'agit ici de vérifier :</p> <ul style="list-style-type: none"> <li>la prise en compte des plans de production nominaux de l'OPS,</li> <li>la prise en compte de la parallélisation de l'OPS,</li> <li>que les données statiques lues en début de traitement sont accessibles par les algorithmes multithreadés,</li> <li>que les données dynamiques lues en début de granule sont accessibles par les algorithmes multithreadés,</li> <li>que les données dynamiques créées par les algorithmes sont accessibles et écrites dans les produits en sortie.</li> </ul>		
<p><b>Conditions initiales requises :</b></p> <p>Au préalable à ce test, le test INT-ALGO-N-01 a été mené de manière à intégrer les chaînes de traitement dans un contexte mono-thread., De plus l'ensemble de l'architecture multi-thread générique du SD a été testée unitairement .</p> <p>Des données statiques et dynamiques cohérentes sont disponibles (données de validation V0)</p>		
Description :	Résultat attendu :	Interfaces testées :
1. Lancer le SD en mode prototype avec deux threads de calcul.	2 processus sont alors présents : le SD et le simulateur de WOM qui permet de piloter le SD sans passer par l'architecture multi-process de l'OPS. Lors de cette phase, les données statiques doivent être lues et stockées en interne.	[f_ParametresAutres] [f_BanqueSpectrale] [f_ParametreStables] [f_Contexte] [f_AlgoTrace]
2. Emettre à l'aide du simulateur WOM un ordre de traitement 0-1C : <ul style="list-style-type: none"> <li>initialisation des données dynamiques,</li> <li>étape 1,</li> <li>étape 2,</li> <li>étape 3,</li> <li>étape 4,</li> <li>écriture des données.</li> </ul>	<p>Vérifier par échantillonnage que les données lues sont correctement reçues par les algorithmes de la première étape dans un contexte multithread.</p> <p>Vérifier par échantillonnage que les données générées par la première étape sont accessibles par la deuxième dans un contexte multithread.</p> <p>Vérifier par échantillonnage que les données générées par les étapes de calcul sont prises en compte par les fonctions d'écriture de donnée dans un contexte multithread</p> <p>NB : on pourra se baser ici sur des traces spécifiques produites par le logiciel et sur un dump des produits en sortie.</p> <p>Vérifier que le déroulement de la chaîne correspond au plan de production (phases parallélisées et phase de regroupement)</p>	[f_ParametresAutres] [f_ParametreStables] [f_BanqueSpectrale] [f_DonneesAuxiliaires] [f_Contexte] [f_ProductsModels] [f_ProduitIASI-In] [f_AlgoTrace] [Données internes] [f_Contexte] [f_ProduitsIASI-1A,1B,1C] [f_DonneesTechnoVerif]

3. Arrêter le SD.		
-------------------	--	--



### 5.1.3.INT-STRUC-N-01

Fiche de Test	INT-STRUC-N-01	OPS
<b>But du test :</b>  <i>Tester la couche de communication, le démarrage et l'arrêt de l'OPS.</i>		
<b>Conditions initiales requises :</b>  Initialiser la configuration afin que seuls les process MSGS, TES , JDBS et MP soient lancés.  Initialiser la configuration afin que l'OPS démarre en standalone.  Initialiser la période d'envoi de la HKTM à 30s		
Description :	Résultat attendu :	Interfaces testées :
1. Lancer l'OPS en mode standalone	Vérifier que les processus MSGS, TES, JDBS et MP sont lancés.  Vérifier que des messages LogEvent ont été émis.  Vérifier que des statuts HKTM ont été émis.	[mcs_HKTM] [m_LogEvent] [pgf_LogEvent]
2. Attendre au moins 30 s	Vérifier que l'ensemble des statuts de HKTM sont émis toutes les 30s.  Vérifier que les status de type ressource font l'objet d'un LogEvent.	[m_InitTimer] [m_EvtTimer]
3. Emettre une commande de type Stop	Vérifier que tous les processus sont arrêtés  Vérifier que l'arrêt est tracé par des LogEvent. Vérifier le compte rendu d'exécution de la commande Stop.	[pgf_Cmd] [pgf_CmdTM] [m_LogEvent] [pgf_LogEvent] [mcs_HKTM] [m_FinTES] [m_FinJDBS]

### 5.1.4.INT-STRUC-D-02

Fiche de Test	INT-STRUC-D-02	OPS
<b>But du test :</b>  <i>Tester la mort d'un processus.</i>		
<b>Conditions initiales requises :</b>  Initialiser la configuration afin que seuls les process MSGS, TES , JDBS et MP soient lancés.  Initialiser la configuration afin que l'OPS démarre en standalone.  Initialiser la période d'envoi de la HKTM à 30s		
Description :	Résultat attendu :	Interfaces testées :
1. Lancer l'OPS en mode standalone.		[pgf_CmdTM] [mcs_HKTM] [m_LogEvent] [pgf_LogEvent]
2. Tuer le processus TES	Vérifier que le MP détecte la mort du TES et loggue un logEvent. Vérifier que l'OPS s'arrête complètement. Vérifier d'un status Operational readiness de valeur Unrecoverable error est émis.	[m_Mort]

### 5.1.5.INT-APPL-N-01

Fiche de Test	INT-APPL-N-01	OPS
<b>But du test :</b>  <i>Tester les interfaces entre les process SD et WOM</i>		
<b>Conditions initiales requises :</b>  Initialiser la configuration afin que seuls les process MSGS, TES , JDBS, SD et WOM soient lancés.  Initialiser la configuration afin que les processus démarrent en standalone.  Initialiser la période d'envoi du pipeline status par le SD à 30s		
Description :	Résultat attendu :	Interfaces testées :
1. Lancer le simulateur de MP et les processus à intégrer en mode standalone.	Vérifier que les processus MSGS, TES, JDBS, SD et WOM sont lancés.	[m_LogEvent] [pgf_LogEvent]
2. Attendre la fin d'initialisation du SD	Vérifier l'émission par le WOM d'un compte rendu d'exécution de commande et du LogEvent associé. Vérifier l'émission de l'Operational Status	[m_CmdState] [m_FinInit] [m_LogEvent] [pgf_LogEvent] [m_OperationalStatus]
3. Déposer un work Order sous le répertoire d'entrée de l'OPS (ce work order est le dernier du dump : type last).	Le work order est disponible pour être traité.	[f_WorkOrder]
4. Emettre une commande Step	Le SD simule le traitement du work order. Vérifier les logEvent qui tracent l'exécution des traitements.	[m_ProcessingCtrl] [m_Traitement]
5. Vérifier que périodiquement le pipeline status est renvoyé à destination du MP.		[m_InitTimer] [m_EvtTimer] [m_PipelineStatus]

6. Attendre la fin de la simulation du traitement.	Un logEvent de fin de traitement est émis au JDB. Vérifier la mise à disposition des produits simulés, du fichier contexte et des données Technologiques/Vérification. Vérifier la génération du report associé. Vérifier l'émission de l'operational Status : Idle. Vérifier l'émission des stages TM associés aux produits et reports générés. Vérifier l'arrêt du WOM sur le traitement du dernier WO du dump	[m_FinTraitement] [f_ProduitsIASI1A,1B,1C] [f_Contexte] [f_DonnéesTechnoVerif] [f_Report] [m_StageTM] [m_FinDump] [m_Mort] [m_OperationalStatus]
7. Arrêter les autres processus		

## 5.1.6.INT\_APPL\_N02

Fiche de Test	INT-APPL-N-02	OPS
<b>But du test :</b>  <i>Tester les interfaces de Processing Control entre les process SD et WOM</i>		
<b>Conditions initiales requises :</b>  Initialiser la configuration afin que seuls les process MSGS, TES, JDBS, SD et WOM soient lancés.  Initialiser la configuration afin que les processus démarrent en standalone.  Initialiser la période d'envoi du pipeline status par le SD à 30s		
Description :	Résultat attendu :	Interfaces testées :
1. Lancer le simulateur de MP et les processus à intégrer en mode standalone	Vérifier que les processus MSGS, TES, JDBS, SD et WOM sont lancés.	[m_LogEvent] [pgf_LogEvent]
2. Déposer un work order (le dernier du dump : type last) sous le répertoire d'entrée de l'OPS.	Le work order est disponible pour être traité.	[f_WorkOrder]
3. Lancer le simulateur de MP et émettre une commande Step	Le SD simule le traitement du work order. Vérifier les logEvent qui tracent l'exécution des traitements.	[m_ProcessingCtrl] [m_Traitement]
4. Émettre une commande Break.	Un logEvent de fin de traitement est émis au JDB. Vérifier l'émission de l'operational Status : Idle.	[m_ProcessingCtrl] [m_ArretTraitement] [m_FinTraitement] [m_OperationalStatus]
5. Émettre une commande Step	Le SD simule le traitement du work order. Vérifier les logEvent qui tracent l'exécution des traitements.	[m_ProcessingCtrl] [m_Traitement]
6. Émettre une commande Suspend	Le traitement est suspendu. Un Log Event trace cet événement.	[m_Suspend]
7. Émettre une commande Resume	Le traitement est relancé. Un Log Event trace cet événement.	[m_Continue]

8. Attendre la fin de la simulation du traitement	<p>Un logEvent de fin de traitement est émis au JDB.</p> <p>Vérifier la mise à disposition des produits simulés, du fichier contexte et des données Technologiques/Vérification.</p> <p>Vérifier la génération du report associé.</p> <p>Vérifier l'émission de l'operational Status : Idle.</p> <p>Vérifier l'émission des stages TM associés aux produits et reports générés.</p> <p>Vérifier l'arrêt du WOM sur le traitement du dernier WO du dump</p>	<p>[m_FinInit]</p> <p>[f_ProduitsIASI1A,1B,1C]</p> <p>[f_Contexte]</p> <p>[f_DonnéesTechnoVerif]</p> <p>[f_Report]</p> <p>[m_StageTM]</p> <p>[m_FinDump]</p> <p>[m_Mort]</p> <p>[m_OperationalStatus]</p>
9. Arrêter les autres processus		

### 5.1.7.INT-OPS-N-01

Fiche de Test	INT-OPS-N-01	OPS
<b>But du test :</b>  <i>Tester le démarrage et l'arrêt du logiciel OPS.</i>		
<b>Conditions initiales requises :</b>  Initialiser la configuration afin que tous les process de l'OPS soient lancés.  Initialiser la configuration afin que l'OPS démarre en standalone.  Initialiser la période d'envoi du pipeline status par le SD et de l'émission de la HKTM à 30s		
Description :	Résultat attendu :	Interfaces testées :
1. Lancer l'OPS en mode standalone	Vérifier que tous les processus sont lancés.	[pgf_CmdTM] [mcs_HKTM] [m_LogEvent] [pgf_LogEvent]
2. Emettre une commande de type Stop	Vérifier que tous les processus sont arrêtés..	[m_FinTES] [m_FinJDBS] [m_FinSD] [m_FinWOM] [m_Mort]

## 5.2.TESTS DE VALIDATION

### 5.2.1.GEN-OPS-N-01

Test Sheet	GEN-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the software generation procedures :</p> <ul style="list-style-type: none"> <li>• build the executable software from the source code,</li> <li>• generation of the installation kit.</li> </ul>		
<p><b><u>Pre requisites :</u></b></p> <p>The following furnitures are needed :</p> <ul style="list-style-type: none"> <li>• the CDROM where the current OPS software version extracted of the configuration management is stored,</li> <li>• the Transfert Document [DA120].</li> </ul>		
Description :	Expected Result :	Tested Requirement:
1. Connect as generation username iasi_1mg on the PGE framework machine.		
2. Follow the generation instructions listed in the Transfert Document after creating the generation directory.  <b>mkdir -p \$HOME/Recette_Vx/OPS/Generation</b>	<p>Verify that the following file types are delivered on the CDROM:</p> <ul style="list-style-type: none"> <li>• source code,</li> <li>• configuration file,</li> <li>• installation scripts,</li> <li>• test tools,</li> <li>• a context file for first launch (« initialisation à froid ») [not delivered in V0].</li> </ul> <p>Verify that the expected generation log messages are produced.</p> <p>The installation kit is generated and stored onto a 4mm cartridge (DDS4 format) support.</p>	<p>[E_CTX4] [E_INI5] [SYS-0150]</p>



3. The DDC includes a dedicated chapter for the generation and installation procedures. <b>vi DDC_OPS_&lt;version&gt;.txt</b>	Check that these informations are included in the DDC.	[SYS-0150]
--	--	------------

## 5.2.2.INST-OPS-N-01

Test Sheet	INST-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to install the OPS software onto the target computer.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The following furnitures are needed :</p> <ul style="list-style-type: none"> <li>the installation kit stored on a 4 mm cartridge (DDS4 compatible) issued from the generation procedure : GEN-OPS-N-01 test,</li> <li>the installation manual [DA114].</li> </ul> <p>The OPS installation directory tree is available on the target test machine : SIM-OPS-N-01</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connect as iasi_1mg user on the target computer after creating the Installation directory.</p> <p><b>mkdir -p \$HOME/Recette_Vx/OPS/Installation.</b></p>		
<p>2. Follow the installation instructions listed in the Installation Manual (chapter 4).</p>	<p>Verify that the expected installation log messages are produced.</p> <p>Verify that the expected directories and files are available on the installation machine.</p>	<p>[SYS-0110]</p> <p>[SYS-0080 bullet3]</p>
<p>3. Check by inspection that the installation procedure is in accordance with the installation requirements (see [DA18] §6.3).</p>	<p>Verify that the installation procedure is divided into 2 successive steps : the application software installation and the application software configuration.</p>	<p>[SYS-0110]</p>
<p>4.</p>	<p>Verify that the OPS and each executable has its own configuration file (OPS.env, &lt;process_name&gt;.env).</p>	<p>[SYS-0120]</p>

5.	Verify that the OPS delivery includes all the configuration file (*.env, *.cfg, *.cat) and data (cold configuration parameters[not delivered for V0]) for an immediate use. In particular, software configuration is predefined. Verify that all configuration files are designed in an ASCII data structure	[PPS-3.7-090]  [SYS-0730]
6. Edit the OPS shell script :  <b>cd</b> <b>&lt;WORKING_ROOT_DIR&gt;/OPS_versions/OPS_&lt;version&gt;/cmd</b>  <b>more *.sh</b>	Verify that only ksh scripts are used.	[SYS-0130] [SYS-0140]
7. Default software configuration files are installed.  If, it is necessary to update them, follow the configuration instructions described in the Installation Manual.	The OPS is configured, ready for use.	
8. If the validation is not performed on the target platform in CGS environment, update the « env_test.sh » :  <b>cd</b> <b>&lt;WORKING_ROOT_DIR&gt;/OPS_versions/OPS_&lt;version&gt;/tools/ OPStools</b>  <b>vi env_test.sh</b>  in order to initialise environment variables :  • WORK_ROOT [ <u>absolute path</u> ] • MLA_ROOT • PGFSIM_ROOT  Source <i>env_test.sh</i>  <b>./env_test.sh</b>		
9. Initialize the current OPS version  <b>cd \$WORK_DIR</b>  <b>ln -s OPS_versions/OPS_&lt;version&gt; OPS</b>	Verify that the current version is the expected one  <b>ls -l OPS</b>	

### 5.2.3.INST-OPS-N-02

Test Sheet	INST-OPS-N-02	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to install the OPS test data not stored in the configuration management onto the target computer.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The following furnitures are needed :</p> <ul style="list-style-type: none"> <li>the 4mm cartridge where test data are stored,</li> <li>the installation manual [DA114]..</li> </ul> <p>The OPS installation directory tree is available on the target test machine : SIM-OPS-N-01</p>		
Description :	Expected Result :	Tested Requirement:
1. Connect as iasi_1mg user on the target computer.		
2. Extract input test data after creating the temporary directory used for reference data (second cadridge) :  <b>mkdir -p \$HOME/Recette_Vx/tmp</b>  Follow the installation instructions listed in the Installation Manual (see Data and Validation Tools Installation chapter).	Verify that the data are extracted and copied on the target computer.	[PPS-3.6-070]

3. Create links between PGF/MLA Simu directories and PGF/MLA Simu binaries :

```
cd <WORKING_ROOT_DIR>/..
```

```
rmdir PGFSimu [if empty directory exists]
```

```
rmdir MLASimu [if empty directory exists]
```

```
ln -s
```

```
<WORKING_ROOT_DIR>/OPS_versions/OPS_<  
version>/tools/PGFSimu PGFSimu
```

```
ln -s
```

```
<WORKING_ROOT_DIR>/OPS_versions/OPS_<  
version>/tools/MLASimu MLASimu
```

Verify that links are correctly created

## 5.2.4.INST-OPS-N-03

Test Sheet	INST-OPS-N-03	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to install a new version of the OPS software onto the target computer while the previous version is running.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version have been installed : execution of GEN-OPS-N-01 and RUN-OPS-N-01 tests.</p>		
Description :	Expected Result :	Tested Requirement:
1. Open a dtterm and connect as iasi_1mg user on the target computer.		
2. In order to simulate the installation of a new version : in the delivery directory, patch the version file (OPS_<version>/ProdConfig/OPS_version.cf) in order to increment the current version number to the following version.	The version file is updated	
3. Replay step 2 of GEN-OPS-N-01.	A new version is ready.	
4. Install and configure this new version (see step 8 of INST-OPS-N-01).	The following version is installed.	[SYS-0161] [PPF-0040] [PPF-0554] [PPF-0960] [PPS-3.5-040]
5. In iasi_1 dtterm : Stop the OPS.	The OPS is stopped.	
6. In iasi_1mg dtterm Set this new version as the current one (see step 9 of INST-OPS-N-01 and step 3 of INST-OPS-N-02 test).	The new version becomes the current one.	

7. In iasi_1 dtterm : Start the OPS sub-system (see RUN-OPS-N-01)	The new OPS version is running.	
8. In iasi_1 dtterm : Stop the OPS.	The OPS is stopped.	

### 5.2.5.SIM-OPS-N-01

Test Sheet	SIM-OPS-N-01	
<b><u>Purpose :</u></b> The purpose of this test is to create a simulated CGS execution environment.		
<b><u>Pre requisites:</u></b> The following furniture is needed : <ul style="list-style-type: none"><li>the installation kit stored on a 4 mm cartridge (DDS4 compatible) issued from the generation procedure : GEN-OPS-N-01 test.</li></ul>		
Description :	Expected Result :	Tested Requirement:
1. In a dtterm, connect as iasi_1mg user on the target computer.		



## 2. Create the simulated CGS repertory tree :

Extract from the delivered tar file the generation repertory tree procedure :

```
mkdir $HOME/tmp
```

```
mkdir -p
```

```
$HOME/Recette_Vx/OPS/Installation
```

```
mv OPS_Install_<version>.tar.Z
```

```
$HOME/tmp
```

```
cd $HOME/tmp
```

```
zcat OPS_Install_<version>.tar.Z | tar xvf -  
./OPS_<version>/tools/OPStools/CreateTre  
e.sh
```

Execute the generation procedure :

```
OPS_<version>/tools/OPStools/CreateTree.  
sh
```

Enter the existing directory

```
$HOME/Recette_Vx/OPS/Installation
```

Delete the extracted files :

```
cd $HOME
```

```
rm -R tmp/
```

The simulated CGS directory tree is created (see in Run time Directory Tree chapter of Installation Manual [DA114]).

## 5.2.6.SIM-OPS-N-02

Test Sheet		SIM-OPS-N-02	
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to start the CGS simulation environment compound of the MLA and the PGF simulators.</p>			
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>The PGF and MLA simulator are installed on the target computer : the simulators are installed simultaneously with the operational software on the OPS/tools directories.</p> <p>Connected as iasi_1mg, configure the OPS in CGS (see Installation Manual [DA114]).</p> <p>(e.g export CHN_OPS_INSTAL_TYPE="CGS" in file OPS.env)</p>			
Description :		Expected Result :	Tested Requirement:
<p>1. In a dtterm window (connected as iasi_1), launch the PGF simulator :</p> <p><b>cd &lt;WORKING_ROOT_DIR&gt;/../PGFSimu</b></p> <p><b>StartPGFSimulator.sh</b></p> <p><b>&lt;WORKING_ROOT_DIR&gt; N MANUAL</b></p>		The PGF simulator is running.	
<p>2. In the PGFsimulator window, launch the OPS.</p> <p>Execute the first 3 steps of RUN-OPS-N-01 test.</p>		The OPS is ready to process input data in nominal mode.	[E_IFS6 pt5]

<p>3. Connected as iasi_1</p> <p>Copy the OPS input processing data, needed to process the whole granule, in the input directory</p> <pre>cd \$WORK_DIR/OPS/tools/OPStools copyTestData.sh SIM_OPS_N_02 input/aux_data</pre> <p>4. Copy the OPS input processing data, needed to process the first granule, in the input directory</p> <pre>copyTestData.sh SIM_OPS_N_02 input/product_model copyTestData.sh SIM_OPS_N_02 input/unproc_data</pre> <pre>cp \$WORK_DIR/OPS/data//TV/SIM_OPS_N_02/input/IASI_1_wo_001bis \$WORK_DIR/input/.</pre>	<p>Verify with the</p> <pre>ls \$WORK_DIR/input/aux_data</pre> <p>command that all the input aux data are available.</p> <p>Verify with the</p> <pre>ls \$WORK_DIR/input ls \$WORK_DIR/input/product_model ls \$WORK_DIR/input/unproc_data</pre> <p>commands that all the input data are available.</p>	<p>[PPS-3.2-110 pt1,2,3,4,7]</p> <p>[PPS-3.2-110 pt1,2,3,4,7]</p>
<p>5. Enter:</p> <p><b>STEP 2 input/IASI_1_wo_001bis</b></p> <p><b>STOP 3</b></p> <p>The MLA simulator is started in STUB mode. In this mode, the HKTM status and events are stored in files.</p> <p>To display the received HKTM status ; in a ddterm (connected as iasi_1) execute the following commands :</p> <pre>cat &lt;WORKING_ROOT_DIR&gt;/tmp/HKTM*</pre> <p>To display the received events ; in a ddterm (connected as iasi_1) execute the following commands :</p> <pre>cat &lt;WORKING_ROOT_DIR&gt;/tmp/LOG*</pre>	<p>The results are the same as the ones obtained by the GRAN_OPS_N_01 test.</p>	
<p>6. Stop the PGF simulator , enter :</p> <p><b>PGFS_ABORT</b></p>	<p>The PGF simulator is stopped.</p>	

## 5.2.7.RUN-OPS-N-01

Test Sheet	RUN-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to start the OPS.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>The input data are available : execution of INST-OPS-N-03 test</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-02 test.</p>		
Description :	Expected Result :	Tested Requirement:
1. In a dtterm, connect as iasi_1 user on the target computer.		
2. Initialize test environment <b>cd &lt;WORKING_ROOT_DIR&gt;</b> <b>./OPS/tools/OPStools/env_test.sh</b>		
3. If the test is executed in CGS simulated environment, start the CGS simulators see SIM-OPS-N-02 test step 1		

<p>4. Launch the OPS subsystem :</p> <p>Enter start command in the PGFS simulator window:</p> <p><b>START 1 "WORK_DIR" N</b></p> <p><b>with "WORK_DIR" = <u>path to working root directory</u></b></p>	<p>Verify with the</p> <p><b>ps -ef   grep Serveur</b></p> <p><b>ps -ef   grep MainProcess</b></p> <p>commands that the OPS processes are running :</p> <ul style="list-style-type: none"> <li>• MSGS__Serveur ,</li> <li>• MP__MainProcess,</li> <li>• TES__ServeurTemps,</li> <li>• JDBS__Serveur,</li> <li>• SD_FRW__ServeurDonnees,</li> <li>• WOM__ServeurWorkOrder.</li> </ul> <p>Verify that :</p> <ul style="list-style-type: none"> <li>• log events trace that the OPS is started,</li> <li>• the PPF version status which provides the IASI-OPS identity is sent to the MCS,</li> <li>• a readiness status « idle » is sent to the MCS,</li> <li>• a START command acknowledgement CODED is sent to the PGF synchronously with the associated log event to the MCS.</li> </ul>	<p>[SYS-0290]</p> <p>[SYS-0380]</p> <p>[PPF-0770]</p> <p>[PPF-0390pt1]</p> <p>[SYS-0330 category P et O]</p> <p>[SYS-0490]</p> <p>[IF-ALGO-4.2-140]</p> <p>[PPS-3.2-030]</p> <p>[IF-ALGO-4.2-100]</p>
<p>5. Stop the OPS : send a STOP command in the PGF simulator window :</p> <p><b>STOP 2</b></p>	<p>A START command acknowledgement ABORTED is sent to the PGF synchronously with the associated log event to the MCS in less than 15s.</p> <p>The OPS is stopped in less than 1mn.</p> <p>Verify with the</p> <p><b>ps -ef   grep Serveur</b></p> <p><b>ps -ef   grep MainProcess</b></p> <p>commands that no OPS process is still running.</p>	<p>[PPS-3.2-030]</p> <p>[IF-ALGO-4.2-100]</p> <p>[SYS-0390]</p> <p>[PPF-0690]</p> <p>[PPF-0350]</p> <p>[PPF-0550]</p>
<p>6. Clean the OPS validation environment :</p> <p><b>cd \$WORK_DIR/OPS/tools/OPStools</b></p> <p><b>Clean.sh</b></p>	<p>Verify that : input, to_pgf and tmp directories are purged:</p> <p>ls \$WORK_ROOT_DIR/input</p> <p>ls \$WORK_ROOT_DIR/to_pgf</p> <p>ls \$WORK_ROOT_DIR/tmp</p>	
<p>7. If the test is executed in CGS simulated environment, stop the CGS simulators</p> <p>see SIM-OPS-N-02 test step 6</p>		

## 5.2.8.RUN-OPS-D-02

Test Sheet	RUN-OPS-D-02	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to test the OPS abort meta command processing.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been install successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
1. Launch the OPS : Execute the first 3 steps of the RUN-OPS-N-01 test.	The OPS is ready to process input data.	
2. Send an abort command to the OPS processus <b>kill -15 &lt;TES id&gt; &lt;SD id&gt; &lt;MP id&gt; &lt;JDBS id&gt; &lt;WOM id&gt; &lt;MSGs id&gt;</b>	Verify with the <b>ps -ef   grep MainProcess</b> <b>ps -ef   grep Serveur</b> commands that all the OPS process are stopped. Verify that process are stopped without generating any log events nor HKTM status. <b>cd \$WORK_DIR/tmp</b> <b>cat HKTM*</b> <b>cat LOG*</b>	[SYS-0380] PPS-3.2-030] [IF-ALGO-4.2-100] [PPF-0430]
3. Clean the OPS validation environment : <b>cd \$WORK_DIR/OPS/tools/OPStools</b> <b>Clean.sh</b>	Verify that : input, to_pgf and tmp directories are purged: <b>ls \$WORK_ROOT_DIR/input</b> <b>ls \$WORK_ROOT_DIR/to_pgf</b> <b>ls \$WORK_ROOT_DIR/tmp</b>	

## 5.2.9.RUN-OPS-D-03

Test Sheet	RUN-OPS-D-03	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to test the OPS behaviour in case of missing configuration file.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been install successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p> <p>Connected as iasi_1mg, rename a SD task diagram configuration file :</p> <pre>mv &lt;WORKING_ROOT_DIR&gt;/OPS/config/ModelesActions.cfg &lt;WORKING_ROOT_DIR&gt;/OPS/config/ModelesActions.sav</pre>		
Description :	Expected Result :	Tested Requirement:
1. Launch the OPS : Execute the first 3 steps of RUN-OPS-N-01 test.	The OPS detects that a mandatory configuration file is missing, sends a START command acknowledgement REJECTED and stops.	[SYS-0440]
2. Connected as iasi_1mg, reinitialize the original configuration.		
3. Clean the OPS validation environment : <code>cd \$WORK_DIR/OPS/tools/OPStools</code> <code>Clean.sh</code>	Verify that : input, to_pgf and tmp directories are purged.	

## 5.2.10.CONF-OPS-N-01

Test Sheet	CONF-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the initialisation stage of the IASI L1 processing.</p> <p>Used data set : J1</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
<b>Action :</b>	<b>Verification :</b>	<b>Tested Requirement:</b>



<p>1. Replay the GRAN-OPS-N-04 test</p>	<p>Verify that the STEP command is acknowledged.</p> <p>Verify that the process size (SD) has increased so that it is greater than the total size of all the auxiliary files (spectral database + configuration files) :</p> <p><b>ps -ef   grep Serveur</b></p> <p>Verify that :</p> <ul style="list-style-type: none"> <li>log events trace the initialisation of the static configuration for each file (other, stable, spectral database, context file)</li> </ul> <p><b>cd output</b></p> <p><b>cat LOG*</b></p> <p>the performance of the initialisation stage is compatible with specifications (15 mn)</p> <p>do the difference between the first log message of the OPS and the message of start of first step in TRACE* file</p> <p><b>cd output</b></p> <p><b>cat TRACE*</b></p>	<p>[E_FIC1] [E_FIC2] [E_FIC3] [E_FIC4] [E_INI2] [E_INI3] [E_INI4]  [PPS-3.4-040]</p>
<p>2. Verify the content- of the context file</p>	<p>A dump of the context file is generated to show its content. The size of the context file is verified.</p>	

## 5.2.11.CONF-OPS-D-02

Test Sheet	CONF-OPS-D-02	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the initialisation stage of the IASI L1 processing and to simulate a modification of the configuration with the absence of the spectral database file.</p> <p>Used data set :J11 bis</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Launch the OPS :</p> <p>Execute the first 3 steps of RUN-OPS-N-01 test.</p>	The OPS is ready to process input data in nominal mode.	
<p>2. Connected as iasi_1 then go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/CONF_OPS_D_02</code></p> <p>then execute the test shell :</p> <p><b><code>./lancetestbis.sh</code></b></p> <p>This shell executes automaticaly the following steps, in italic style</p>		

<p>3. <i>Copy(using symbolic links) all data needed to process both granules in the directories</i>  <i>\$WORK_DIR/input/aux_data</i>  <i>\$WORK_DIR/input/unproc_data</i>  <i>\$WORK_DIR/input/product_model</i></p>	<p>Data are copied in  <b>\$WORK_DIR/input/aux_data</b>  <b>\$WORK_DIR/input/unproc_data</b>  <b>\$WORK_DIR/input/product_model</b></p> <p>A screen message confirms the operation :  <i>mise en place des données auxiliaires</i>  <i>Copy SUCCESS</i>  <i>END</i>  <i>mise en place des product model</i>  <i>Copy SUCCESS</i>  <i>END</i><i>mise en place des données d'entrée</i>  <i>Copy SUCCESS</i>  <i>END</i></p>	
<p><u>AVHRR product</u> : J1 ( see § 3.5.2)  <u>Other and Stable Parameters , Spectral Database</u> : J11 bis ( see § 3.5.2)  <u>IASI Context file</u> : TV_THA_CTX_1 ( see § 3.5.2)  <u>Auxiliary data</u> : J1 ( see § 3.5.2)</p>		
<p><u>working order</u> : IASI_1_wo_001  <u>IASI L0 data</u> : J6 ( see § 3.5.2) ( first granule)  <u>Product model</u> : J11bis ( see § 3.5.2) ( 5 files)</p>		
<p>4. <i>Send then a STEP command to the OPS subsystem in order to execute WorkingOrder n°1 :</i>  <i>STEP 1 input/IASI_1_wo_001</i></p> <p><i>Wait for the processing end.</i></p>	<p>Verify that the STEP command is sent and acknowledged by a screen message :  ACK STEP 1 0  Verify that WorkingOrder 1 as begun :  ACK START 1 0  Verify after step 7 that :</p> <ul style="list-style-type: none"> <li>log events trace the initialisation of the static configuration ,</li> <li>log events trace the correct execution of the order and the L1 products generation</li> </ul> <p>Verify the process terminates correctly =&gt; screen message ACK START 1 1</p>	
<p><u>working order</u> : and IASI_1_wo_002  <u>IASI L0 data</u> : J6 ( see § 3.5.2) ( second granule)  <u>Product model</u> : J11bis ( see § 3.5.2) ( 5 files)</p>		

<p>5. <i>Send then a STEP command to the OPS subsystem in order to execute WorkingOrder n°1 :</i></p> <p><b>STEP 2 input/IASI_1_wo_002</b></p> <p><i>Wait for the processing end.</i></p>	<p>Verify that the STEP command is sent and acknowledged by a screen message :</p> <p>ACK STEP 2 0</p> <p>Verify that WorkingOrder 2 as begun :</p> <p>ACK START 2 0</p> <p>Verify after step 7 that :</p> <ul style="list-style-type: none"> <li>• log events trace a warning indicating that PTSI and spectral database have changed</li> <li>• A warning notifying the absence of spectral database is emitted;</li> <li>• The processing chain continues and generate IASI L1 products.</li> </ul> <p>Verify the process terminates correctly =&gt; screen message ACK START 2 1</p>	[PPF-0610pt3]
<p>6. <i>Clean the OPS validation environment by executing the shell :</i></p> <p><b>\$WORK_DIR/OPS/tools/OPStools/Clean.sh</b></p>	<p>Verify that : input, to_pgf and tmp directories are purged.</p>	
<p>7. <i>Copies LOG files, TRACE files and results in the output directory</i></p> <p><b>\$WORK_DIR/OPS/data/TV/CONF_OPS_D_02/output</b></p>	<p>Verify all products, LOG files and TRACE files are present in the directory :</p> <ul style="list-style-type: none"> <li>• 2 products ENG</li> <li>• 2 products VER</li> <li>• 2 products 1A</li> <li>• 2 products 1B</li> <li>• 2 products 1C</li> </ul> <p>Edit TRACE_xxxxxxxx and LOG_xxxxxxxx process through all verifications needed by previous steps.</p>	

**5.2.12.CONF-OPS-D-03**

Test Sheet	CONF-OPS-D-03	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the initialisation stage of the IASI L1 processing and to simulate a modification of the configuration with the absence of the stable configuration file</p> <p>Used data set : J11quar</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Launch the OPS :</p> <p>Execute the first 3 steps of RUN-OPS-N-01 test.</p>	The OPS is ready to process input data in nominal mode.	
<p>2. Connected as iasi_1 then go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/CONF_OPS_D_03</code></p> <p>then execute the test shell :</p> <p><b><code>./lancetestbis.sh</code></b></p> <p>This shell executes automatically the following steps, in italic style</p>		

<p>3. <i>Copy(using symbolic links) all data needed to process both granules in the directories</i>  <i>\$WORK_DIR/input/aux_data</i>  <i>\$WORK_DIR/input/unproc_data</i>  <i>\$WORK_DIR/input/product_model</i></p>	<p>Data are copied in  <b>\$WORK_DIR/input/aux_data</b>  <b>\$WORK_DIR/input/unproc_data</b>  <b>\$WORK_DIR/input/product_model</b></p> <p>A screen message confirms the operation :  <i>mise en place des données auxiliaires</i>  <i>Copy SUCCESS</i>  <i>END</i>  <i>mise en place des product model</i>  <i>Copy SUCCESS</i>  <i>END</i><i>mise en place des données d'entrée</i>  <i>Copy SUCCESS</i>  <i>END</i></p>	
<p><u>working order</u> : IASI_1_wo_001  <u>IASI L0 data</u> : J6 ( see § 3.5.2) ( first granule)  <u>Product model</u> : J11quar ( see § 3.5.2) ( 5 files)</p> <p><u>AVHRR product</u> : J1 ( see § 3.5.2)  <u>Other and Stable Parameters , Spectral Database</u> : J11 quar ( see § 3.5.2)  <u>IASI Context file</u> : TV_THA_CTX_1 ( see § 3.5.2)  <u>Auxiliary data</u> : J1 ( see § 3.5.2)</p>		
<p>4. <i>Send then a STEP command to the OPS subsystem in order to execute WorkingOrder n°1 :</i>  <b>STEP 1 input/IASI_1_wo_001</b></p> <p>1. <i>Wait for the processing end.</i></p>	<p>Verify that the STEP command is sent and acknowledged by a screen message :  <b>ACK STEP 1 0</b></p> <p>Verify that WorkingOrder 1 as begun :  <b>ACK START 1 0</b></p> <p>Verify after step 7 that :</p> <ul style="list-style-type: none"> <li>log events trace the initialisation of the static configuration ,</li> <li>log events trace the correct execution of the order and the L1 products generation</li> </ul> <p>Verify the process terminates correctly =&gt; screen message <b>ACK START 1 1</b></p>	
<p><u>working order</u> : IASI_1_wo_002  <u>IASI L0 data</u> : J6 ( see § 3.5.2) ( second granule)  <u>Product model</u> : J11quar ( see § 3.5.2) ( 5 files)</p>		

<p>5. <i>Send then a STEP command to the OPS subsystem in order to execute WorkingOrder n°1 :</i></p> <p><b>STEP 2 input/IASI_1_wo_002</b></p> <p><i>Wait for the processing end.</i></p> <p>6. <i>Clean the OPS validation environment by executing the shell :</i></p> <p><b>\$WORK_DIR/OPS/tools/OPStools/Clean.sh</b></p>	<p>Verify that the STEP command is sent and acknowledged by a screen message :</p> <p>ACK STEP 2 0</p> <p>Verify that WorkingOrder 2 as begun :</p> <p>ACK START 2 0</p> <p>Verify after step 7 that :</p> <ul style="list-style-type: none"> <li>• log events trace a warning indicating that PTSI and stable configuration file have changed</li> <li>• A warning notifying the absence of stable configuration file is emitted</li> <li>• The processing chain continues and generate IASI L1 products.</li> </ul> <p>Verify the process terminates correctly =&gt; screen message ACK START 2 1</p> <p>Verify that : input, to_pgf and tmp directories are purged.</p>	[PPF-0610pt3]
<p>7. <i>Copies LOG files, TRACE files and results in the output directory</i></p> <p><b>\$WORK_DIR/OPS/data/TV/CONF_OPS_D_02/output</b></p>	<p>Verify all products, LOG files and TRACE files are present in the directory :</p> <ul style="list-style-type: none"> <li>• 2 products ENG</li> <li>• 2 products VER</li> <li>• 2 products 1A</li> <li>• 2 products 1B</li> <li>• 2 products 1C</li> </ul> <p>Edit TRACE_xxxxxxxx and LOG_xxxxxxxx process through all verifications needed by previous steps.</p>	

### 5.2.13.CONF-OPS-D-04

Test Sheet	CONF-OPS-D-04	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the initialisation stage of the IASI L1 processing and to simulate a modification of the configuration with the absence of the other configuration file.</p> <p>Used data set : J11 cinq</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1 .Launch the OPS :</p> <p>Execute the first 3 steps of RUN-OPS-N-01 test.</p>	The OPS is ready to process input data in nominal mode.	
<p>2 Connected as iasi_1 then go in the test specific directory :</p> <p>cd \$WORK_DIR/OPS/data/TV/CONF_OPS_D_04</p> <p>then execute the test shell :</p> <p><b>./lancetestbis.sh</b></p> <p>This shell executes automaticaly the followig steps, in italic style</p>		



<p>3 <i>Copy(using symbolic links) all data needed to process both granules in the directories</i>  <i>\$WORK_DIR/input/aux_data</i>  <i>\$WORK_DIR/input/unproc_data</i>  <i>\$WORK_DIR/input/product_model</i></p>	<p>Data are copied in  <b>\$WORK_DIR/input/aux_data</b>  <b>\$WORK_DIR/input/unproc_data</b>  <b>\$WORK_DIR/input/product_model</b></p> <p>A screen message confirms the operation :  <i>mise en place des données auxiliaires</i>  <i>Copy SUCCESS</i>  <i>END</i>  <i>mise en place des product model</i>  <i>Copy SUCCESS</i>  <i>END</i><i>mise en place des données d'entrée</i>  <i>Copy SUCCESS</i>  <i>END</i></p>	
<p><u>AVHRR product</u> : J1 ( see § 3.5.2)  <u>Other and Stable Parameters , Spectral Database</u> : J11 cinq ( see § 3.5.2)  <u>IASI Context file</u> : TV_THA_CTX_1 ( see § 3.5.2)  <u>Auxiliary data</u> : J1 ( see § 3.5.2)</p>		
<p><u>working order</u> : IASI_1_wo_001bis  <u>IASI L0 data</u> : J6 ( see § 3.5.2) ( first granule)  <u>Product model</u> : J11cinq ( see § 3.5.2) ( 5 files)</p>		
<p>4 <i>Send then a STEP command to the OPS subsystem in order to execute WorkingOrder n°1 :</i>  <i>STEP 1 input/IASI_1_wo_001bis</i></p> <p><i>Wait for the processing end.</i></p>	<p>Verify that the STEP command is sent and acknowledged by a screen message :  ACK STEP 1 0  Verify that WorkingOrder 1 as begun :  ACK START 1 0  Verify after step 7 that :</p> <ul style="list-style-type: none"> <li>log events trace the initialisation of the static configuration ,</li> <li>log events trace the correct execution of the order and the L1 products generation</li> <li>Verify the process terminates correctly =&gt; screen message ACK START 1 1</li> </ul>	
<p><u>working order</u> : and IASI_1_wo_002bis  <u>IASI L0 data</u> : J4 ( see § 3.5.2) ( second granule)  <u>Product model</u> : J4 ( see § 3.5.2) ( 5 files)</p>		

<p>5. Send then a STEP command to the OPS subsystem in order to execute WorkingOrder n°2 :</p> <p><b>STEP 2 input/IASI_1_wo_002bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is sent and acknowledged by a screen message :</p> <p>ACK STEP 2 0</p> <p>Verify that WorkingOrder 2 as begun :</p> <p>ACK START 2 0</p> <p>Verify that WorkingOrder 2 generates an error :</p> <p>ACK STEP 2 3</p> <p>Then the OPS stops on error:</p> <p>ACK STEP 2 3</p> <p>Verify after step 7 that the TRACE file contains the error message :</p> <ul style="list-style-type: none"> <li>• AUX DATA : NO FILE FOUND WITH PATTERN BRD</li> <li>• THE SD PROCESSUS IS STOPPED ON ERROR</li> </ul>	[PPF-0610pt3]
<p>6. Clean the OPS validation environment by executing the shell :</p> <p><b>\$WORK_DIR/OPS/tools/OPStools/Clean.sh</b></p>	<p>Verify that : input, to_pgf and tmp directories are purged.</p>	
<p>7. Copies LOG files, TRACE files and results in the output directory</p> <p><b>\$WORK_DIR/OPS/data/TV/CONF_OPS_D_02/output</b></p>	<p>Verify all products, LOG files and TRACE files are present in the directory :</p> <ul style="list-style-type: none"> <li>• 1 products ENG</li> <li>• 1 products VER</li> <li>• 1 products 1A</li> <li>• 1 products 1B</li> <li>• 1 products 1C</li> </ul> <p>Edit TRACE_xxxxxxxx and LOG_xxxxxxxx process through all verifications needed by previous steps.</p>	

## 5.2.14.CONF-OPS-D-05

Test Sheet	CONF-OPS-D-05	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the initialisation stage of the IASI L1 processing and to validate the checking of the version of the context</p> <p>Used data set : J11ter</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Launch the OPS :</p> <p>Execute the first 3 steps of RUN-OPS-N-01 test.</p>	The OPS is ready to process input data in nominal mode.	
<p>2. Connected as iasi_1 then go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/CONF_OPS_D_05</code></p> <p>then execute the test shell :</p> <p><b><code>./lancetestbis.sh</code></b></p> <p>This shell executes automatically the following steps, in italic style</p>		

<p>3. <i>Copy(using symbolic links) all data needed to process both granules in the directories</i>  <i>\$WORK_DIR/input/aux_data</i>  <i>\$WORK_DIR/input/unproc_data</i>  <i>\$WORK_DIR/input/product_model</i></p>	<p>Data are copied in  <b>\$WORK_DIR/input/aux_data</b>  <b>\$WORK_DIR/input/unproc_data</b>  <b>\$WORK_DIR/input/product_model</b></p> <p>A screen message confirms the operation :  <i>mise en place des données auxiliaires</i>  <i>Copy SUCCESS</i>  <i>END</i>  <i>mise en place des product model</i>  <i>Copy SUCCESS</i>  <i>END</i><i>mise en place des données d'entrée</i>  <i>Copy SUCCESS</i>  <i>END</i></p>	
<p><u>AVHRR product</u> : J1 ( see § 3.5.2)  <u>Other and Stable Parameters , Spectral Database</u> : J1 ( see § 3.5.2)  <u>IASI Context file</u> : J11ter ( see § 3.5.2)  <u>Auxiliary data</u> : J1 ( see § 3.5.2)</p>		
<p><u>working order</u> : IASI_1_wo_001bis  <u>IASI L0 data</u> : J1 ( see § 3.5.2) ( first granule)  <u>Product model</u> : J1 ( see § 3.5.2) ( 5 files)</p>		
<p><i>Send then a STEP command to the OPS subsystem :</i></p> <p><b><i>STEP 1 input/IASI_1_wo_001bis</i></b></p> <p><i>Wait for the processing end.</i></p>	<p>Verify that the STEP command is acknowledged.  Verify that after step 5 :</p> <ul style="list-style-type: none"> <li>log events trace the error during the initialisation stage due to a bad context file ( integrity check) :INVALID CONTEXT FILE VERSION</li> </ul> <p>Processing chain stops</p>	<p>[PPF-0700]  <a href="#">[E_CON3]</a></p>
<p>5. <i>Clean the OPS validation environment :</i></p> <p><b><i>\$WORK_DIR/OPS/tools/OPStools/Clean.sh</i></b></p>	<p>Verify that : input, to_pgf and tmp directories are purged.</p> <p>Verify that no L1 product is generated</p>	

## 5.2.15.CONF-OPS-D-06

Test Sheet	CONF-OPS-D-06	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate. the IASI L1 chain with degraded input ( format version incoherence) leading to a stop of the processing chain</p> <p>Used data set : J11</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. .Launch the OPS :</p> <p>Execute the first 3 steps of RUN-OPS-N-01 test.</p>	The OPS is ready to process input data in nominal mode.	
<p>2. Connected as iasi_1 then go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/CONF_OPS_D_06</code></p> <p>then execute the test shell :</p> <p><b><code>./lancetestbis.sh</code></b></p> <p>This shell executes automaticaly the following steps, in italic style</p>		

<p>3. <i>Copy(using symbolic links) all data needed to process both granules in the directories</i>  <i>\$WORK_DIR/input/aux_data</i>  <i>\$WORK_DIR/input/unproc_data</i>  <i>\$WORK_DIR/input/product_model</i></p>	<p>Data are copied in  <b>\$WORK_DIR/input/aux_data</b>  <b>\$WORK_DIR/input/unproc_data</b>  <b>\$WORK_DIR/input/product_model</b></p> <p>A screen message confirms the operation :  <i>mise en place des données auxiliaires</i>  <i>Copy SUCCESS</i>  <i>END</i>  <i>mise en place des product model</i>  <i>Copy SUCCESS</i>  <i>END</i><i>mise en place des données d'entrée</i>  <i>Copy SUCCESS</i>  <i>END</i></p>	
<p><u>AVHRR product</u> : J1 ( see § 3.5.2)  <u>Other and Stable Parameters , Spectral Database</u> : J11 ( see § 3.5.2)  <u>IASI Context file</u> : J1 (see § 3.5.2) this file is copied from the delivery  <u>Auxiliary data</u> : J1 ( see § 3.5.2)</p>		
<p><u>working order</u> : IASI_1_wo_001bis  <u>IASI L0 data</u> : J1 ( see § 3.5.2)  <u>Product model</u> : J1 ( see § 3.5.2)</p>		
<p>4. <i>Send then a STEP command to the OPS subsystem :</i>  <i>Enter STEP command in the PGF simulator window for all the granules :</i>  <b>STEP 1 input/IASI_1_wo_001bis</b></p> <p><i>Wait for the processing end.</i></p>	<p>Verify that each STEP command is acknowledged.  A logEvent is generated during the initialisation step due to a bad format version  INVALID OTHER CONFIGURATION FILE VERSION  The OPS stops  Verify that no files are produced :</p>	<p>[PPS-3.3-110pt2]  <a href="#">[E_DEG5]</a></p>

## 5.2.16.GRAN-OPS-D-02

Test Sheet	GRAN-OPS-D-02	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to test the OPS behaviour in case of missing input work order.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
1. Launch the OPS : Execute the first 3 steps of RUN-OPS-N-01 test.	The OPS is ready to process input data in nominal mode.	
1. Connected as iasi_1 . You must go in the test specific directory : <code>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_D_02</code> then execute the test shell : <code>./lancetestbis.sh</code>  <b>This shell execute automaticaly the following steps (in italic style)</b>	Verify that all the input aux data are available with the command :  <b>ls \$WORK_DIR/input/aux_data</b>	[PPS-3.2-110 pt1,2,3,4,7]
2. <i>Clean the OPS validation environment :</i> <code>cd \$WORK_DIR/OPS/tools/OPStools</code> <i>Clean.sh</i>	Verify that <b>input</b> , <b>to_pgf</b> and <b>tmp</b> directories are purged by checking the dates.  <code>ls -l \$WORK_ROOT_DIR/tmp</code> <code>ls -l \$WORK_ROOT_DIR/to_pgf</code>	
3. <i>Launch the OPS in standalone mode:</i> <code>../../cmd/StartStandalone.sh 1</code> <code>\$WORK_DIR N</code>		

4. <i>Copy at first, the input auxiliary data, needed to process all granules in the input directory (using symbolic links)</i>	Verify that all the input data are available with the command :  <b>ls \$WORK_DIR/input/aux_data</b>	
<u>AVHRR product</u> : J10 (see § 3.5.2) <u>Other and Stable Parameters , Spectral Database</u> : J1 (see § 3.5.2) <u>IASI Context file</u> : J1 (see § 3.5.2) <u>Auxiliary data</u> : J1 (see § 3.5.2)		
5. <i>Copy the OPS input processing data, needed to process both granules, in the input directory</i>	Verify that all the input data are available with the commands :  <b>ls \$WORK_DIR/input</b> <b>ls \$WORK_DIR/input/product_model</b> <b>ls \$WORK_DIR/input/unproc_data</b>	[PPS-3.2-110 pt1,2,3,4,7]
<u>working order</u> : IASI_1_wo_001bis and IASI_1_wo_002bis <u>IASI L0 data</u> : J10 ( see § 3.5.2) (two granules) <u>Product model</u> : J10 ( see § 3.5.2) (five files)		
2. <i>Process the first granule:</i>  <b>STEP 1 input/IASI_1_wo_001bis</b>  <i>Wait for the processing end.</i>	Verify that the STEP command is acknowledged.  - during the processing from the directory \$WORK_DIR/tmp with the command :  <b>tail -f CMD*</b>  - at the end of the processing from the directory \$WORK_DIR/OPS/data/TV/GRAN_OPS_D_02/o utput with the command :  <b>nedit CMD*</b>  look for ACK STEP 1 0  Verify that the processing chain stops correctly  <b>nedit CMD*</b>  look for ACK STEP 1 1	
3. <i>Send a STEP command to the OPS subsystem :</i>  <b>STEP 2 input/IASI_1_wo_002bis</b>	Verify with the OPS subsystem detect the missing file and warn the MCS by a log event.  Verify that a STEP command acknowledgement ABORTED is sent.  Then, the OPS waits for another command delivery.	[IF-ALGO-4.1-100]



## 5.2.17.GRAN-OPS-N-04

Test Sheet	GRAN-OPS-N-04	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 processing of a whole product in granule mode.</p> <p>Used data set : J1</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p> <p>If necessary uncompress recursively input data ( directory (\$WORK_DIR/../../tmp)</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test specific directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_N_04</pre> <p>then execute the test shell :</p> <pre>./lancetestbis.sh</pre> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>	Verify that : input, to_pgf and tmp directories are purged.	
<p>3. <i>Launch the OPS in standalone mode:</i></p> <pre>../../cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetestbis.sh):</i></p>	The OPS is ready to process input data in nominal mode.	
<p>4. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i></p>	Verify with the <b>ls \$WORK_DIR/input/aux_data</b> commands that all the input data are available :	

AVHRR product : J1 ( see § 3.5.2)

Other and Stable Parameters , Spectral Database : J1 ( see § 3.5.2)

IASI Context file : J1 (see § 3.5.2) this file is copied from the delivery

Auxiliary data : J1 ( see § 3.5.2)

5. *Copy the processing data, needed to process all granule, in the input directory (using symbolic links)*

Verify with the

**ls \$WORK\_DIR/input**

**ls \$WORK\_DIR/input/product\_model**

**ls \$WORK\_DIR/input/unproc\_data**

commands that all the input data are available :

working order : IASI\_1\_wo\_001 to IASI\_1\_wo\_0046

IASI L0 data : J1 ( see § 3.5.2) ( all granule)

Product model : J1 ( see § 3.5.2) (all granule)

6. Send then a STEP command to the OPS subsystem :

*Enter STEP command in the PGF simulator window for all the granules :*

**STEP 1 input/IASI\_1\_wo\_001**

.....

*Wait for the processing end.*

Verify that each STEP command is acknowledged.

(file CMD\_XXXX)

Verify that with the last granule the processing chain stops

(file CMD\_XXXX)

Verify that the predicted files are produced :

- IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated

- Report file :

**cd output**

**ls**

Verify that the report informations and format are valid.

**cat \*.rpt**

Verify that :

- The last granule is processed :

**cat \*46\*.rpt**

- The verification product size

**ls -l \*VER\***

- The PCD is send using a Log Envent

**more LOGxxx**

[\[\[E\\_IFS10\]](#)

[\[E\\_INT9](#)

[\[PPF-0500\]](#)

[\[PPF-0060\]](#)

[\[PPF-0080\]](#)

[\[PPF-0510\]](#)

[\[PPF-0530\]](#)

[\[PPF-0691\]](#)

[\[PPF-0530\]](#)

[\[PPF-0820\]](#)

<p>7. Verify the format- of the product files (manual)</p> <p><b>cd output</b></p> <p><b>TraceProduit “ product file name ”</b></p> <p><b>GRANx</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with a validation tool (TraceProduit) to verify that the output product are in accordance with the GPFS format. (for example , the header and one record of each type).</p> <p><b>more structure.prod</b> ( it shows 1 MPR 3 IPR and several GIADR/VIADR/MDR)</p>	<p><a href="#">[E_IFS9]</a></p> <p><a href="#">[PPF-0010]</a></p> <p><a href="#">[PPF-0030]</a></p> <p><a href="#">[E_IFS2ter]</a></p> <p><a href="#">[E_IFS3]</a></p> <p><a href="#">[PPS-3.2-110 pt 5 et 6]</a></p>
<p>8. Verify the structure and the content - of the verification product (manual)</p> <p><b>cd \$WORK_DIR/input/unproc_data</b></p> <p><b>TraceProduit “ L0 product file name ”</b></p> <p><b>GRAN0 (for the first granule)</b></p> <p><b>cd</b></p> <p><b>\$WORK_DIR/OPS/data/TV/GRAN_OPS_N_04/output</b></p> <p><b>TraceProduit “ verification file name ”</b></p> <p><b>GRANx</b></p>	<p>The verification product is dumped together with the L0 file ( binary extraction of VPA,VPB,VPC,VPD,VPA,AP ISP).</p> <p>the binary frames are compared on one line.( dmp file)</p> <p><b>cmp -l \$WORK_DIR</b>  <b>/input/unproc_data/GRAN0/MDR1lum/APVP2.</b>  <b>dmp</b>  <b>\$WORK_DIR/OPS/data/TV/GRAN_OPS_N_04/out</b>  <b>put/ GRANx/MDR_VER_1.dmp</b></p> <p>No differences must appear</p>	<p><a href="#">[[E_IFS2ter]</a></p> <p><a href="#">[E_IFS3]</a></p>

## 5.2.18.GRAN-OPS-N-05

Test Sheet	GRAN-OPS-N-05	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate. the IASI L1 processing of a whole product in granule mode with a specific datation</p> <p>Used data set : J1ter</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_N_05</code> then execute the test shell :</p> <p><code>./lancetestbis.sh</code></p> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <p>Connected as iasi_1 . You must go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/tools/OPStools</code></p> <p><i>Clean.sh</i></p>	Verify that : input, to_pgf and tmp directories are purged.	
<p>3. Launch the OPS :</p> <p><code>./.././../cmd/StartStandAlone.sh 1</code>  <code>\$WORK_DIR N</code> (this step is included in the shell lancetestbis.sh):</p>	The OPS is ready to process input data in nominal mode.	

<p>4. <i>Copy at first, the input auxiliary data, needed to process both granules in the input directory (using symbolic links)</i></p>	<p>Verify with the</p> <p><b>ls \$WORK_DIR/input/aux_data</b></p> <p>commands that all the input data are available :</p>	
<p><u>AVHRR product</u> : J1 ( see § 3.5.2)</p> <p><u>Other and Stable Parameters , Spectral Database</u> : J1 ( see § 3.5.2)</p> <p><u>IASI Context file</u> : J1 (see § 3.5.2) this file is copied from the delivery</p> <p><u>Auxiliary data</u> : J1 ( see § 3.5.2)</p>		
<p>5. <i>Copy the processing data, needed to process bote granules, in the input directory (using symbolic links)</i></p>	<p>Verify with the</p> <p><b>ls \$WORK_DIR/input</b></p> <p><b>ls \$WORK_DIR/input/product_model</b></p> <p><b>ls \$WORK_DIR/input/unproc_data</b></p> <p>commands that all the input data are available :</p>	
<p><u>working order</u> : IASI_1_wo_001 IASI_1_wo_002</p> <p><u>IASI L0 data</u> : J1ter ( see § 3.5.2) (all granule)</p> <p><u>Product model</u> : J1ter ( see § 3.5.2) (all granule)</p>		
<p>6. <i>Send then a STEP command to the OPS subsystem :</i></p> <p><i>Enter STEP command in the PGF simulator window for all the granules :</i></p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p>.....</p> <p><i>Wait for the processing end.</i></p>	<p>Verify that each STEP command is acknowledged.</p> <p>Verify that with the last granule the processing chain stops</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> <p><b>cd output</b></p> <p><b>ls *</b></p> <p><b>cat *.rpt</b></p>	<p><a href="#">[[PPS-3.6-030]</a></p> <p><a href="#">[PPS-3.6-050]</a></p>
<p>7. Verify the datation- of the product files (manual)</p> <p><b>cd output</b></p> <p><b>TraceProduit “ product file name ”</b></p> <p><b>GRANx</b></p> <p><b>cd GRANx</b></p>	<p>The first L1C Product file is dumped with. A validation tool to verifie that the output product are in accordance with the GPFS format. (for exemple , the MPHR and the first MDR of each product)</p> <p><b>more structure.prod</b></p> <p><b>more MPHR.tra</b></p> <p><b>more MDR1.tra</b></p> <p><b>the date written in the product must be 30 second upper than those in the first granule of J1 product. ( same files in directory .././references/GRAN1/)</b></p>	

## 5.2.19.GRAN-OPS-N-06

Test Sheet	GRAN-OPS-N-06	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 processing of an intermediate granule using an intermediate context file generated before (backlog processing).</p> <p>Used data set : J3</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test specific directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_N_065</pre> <p>then execute the test shell :</p> <pre>./lancetestbis.sh</pre> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <p>Connected as iasi_1 . You must go in the test specific directory :</p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>	Verify that : input, to_pgf and tmp directories are purged.	
<p>3. Launch the OPS :</p> <pre>./.../cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetestbis.sh):</i></p>	The OPS is ready to process input data in nominal mode.	

<u>AVHRR product</u> : J1 ( see § 3.5.2) <u>Other and Stable Parameters , Spectral Database</u> : J1 ( see § 3.5.2) <u>IASI Context file</u> : J3 (see § 3.5.2) this file is copied from the delivery <u>Auxiliary data</u> : J1 ( see § 3.5.2)		
4. Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)	Verify with the <b>ls \$WORK_DIR/input</b> <b>ls \$WORK_DIR/input/product_model</b> <b>ls \$WORK_DIR/input/unproc_data</b> commands that all the input data are available :	
<u>working order</u> : IASI_1_wo_003 <u>IASI L0 data</u> : J3 ( see § 3.5.2) (third granules of J1) <u>Product model</u> : J1 ( see § 3.5.2) (with backlog mode)		
5. Send then a STEP command to the OPS subsystem : Enter STEP command for all the granules : <b>STEP 3 input/IASI_1_wo_003</b> Enter STOP command after the end of the processing: <b>STOP 1</b>  Wait for the processing end.	Verify that each STEP command is acknowledged. Verify that with the last granule the processing chain stops Verify that the predicted files are produced : <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> Verify that the report informations and format are valid. <b>cd output</b> <b>ls *</b> <b>cat *.rpt</b> Pour le cas particulier où p=q=1, BibMario n'activera	<a href="#">[E_CON1]</a> <a href="#">PPS-3.2-420pt2 et 3]</a> <a href="#">[PPS-3.2-440]</a> [PPF-0630pt3] [PPS-3.2-320] [PPF-0840] [PPF-0850 pt1] [PPF-0880] <a href="#">[PPS-3.6-030]</a>
6. Verify the format- of the context file (manual) <b>cd output</b> <b>TraceContexte "context file name "</b> <b>ctx.tra</b>  <b>more ctx.tra</b>  <b>more ../references/ ctx.tra</b>	The context file is dumped to verify the content and to compare with the context generated in normal mode (GRAN-OPS_N_04) which is dumped in reference directory ( file ctxref.tra)  Few differences are expected (date ...)	<a href="#">[E_CON2]</a> <a href="#">[E_CON4]</a>

<p>7. Compare the content- of the product files</p> <p><b>cd references</b></p> <p><b>TraceProduit “ 1C reference product file name ” GRANx</b></p> <p>(first and second MDR (image and spectra only)</p> <p><b>cd output</b></p> <p><b>TraceProduit “ 1C product file name ” GRANx</b></p> <p>(first and second MDR (image and spectra only)</p> <p><b>cd GRANx</b></p> <p><b>compfic MDRx.tra</b>  <b>.././references/GRANx/MDRx</b>  <b>MDRx.tra.diff 0.0000001</b></p>	<p>The previous files can be compared with files generated in GRAN-OPS-N-04 test using the “compfic” tool</p> <p>The tool must assume no differences with à fine absolute precision ( e.g: 0,0000001) (spectra and images) ( key word NBDIFFBALEUR) in file <b>MDRx.tra.diff.</b></p>	<p><a href="#">[PPF-0820]</a></p>
--	---	-----------------------------------



## 5.2.20.GRAN-OPS-N-07

Test Sheet	GRAN-OPS-N-07	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 processing and to simulate a modification of the configuration (other configuration file).</p> <p>Used data set : J1 and J6</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p> <p>If necessary uncompress recursively input data ( directory \$WORK_DIR/../../tmp)</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_N_07</code></p> <p>then execute the test shell :</p> <p><b><code>./lancetestbis.sh</code></b></p> <p>This shell executes automaticaly the followig steps, in italic style</p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <p><code>cd \$WORK_DIR/OPS/tools/OPStools</code></p> <p><i>Clean.sh</i></p>	<p>Verify that : input, to_pgf and tmp directories are purged:</p> <p><b><code>cd \$WORK_DIR</code></b></p> <p><b><code>ls input</code></b></p> <p><b><code>ls to_pgf</code></b></p> <p><b><code>ls tmp</code></b></p> <p><b><code>ls tmp/data</code></b></p>	
<p>3. <i>Copy at first, the input auxiliary data, needed to process the granules in the input directory (using symbolic links)</i></p>	<p>Verify that all the input data are available with the command :</p> <p><b><code>ls \$WORK_DIR/input/aux_data</code></b></p>	

AVHRR product : J1 ( see § 3.5.2)

Other and Stable Parameters , Spectral Database : J1 for the first granule and J6 for the second granule ( see § 3.5.2)

IASI Context file : J1 ( see § 3.5.2)

Auxiliary data : J1 ( see § 3.5.2)

<p>4. <i>Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</i></p>	<p>Verify that all the input data are available with the commands :</p> <p><b>ls \$WORK_DIR/input</b></p> <p><b>ls \$WORK_DIR/input/product_model</b></p> <p><b>ls \$WORK_DIR/input/unproc_data</b></p>	
---	---	--

working order : IASI\_1\_wo\_001bis

IASI L0 data : J6 ( see § 3.5.2) ( first granule)

Product model : J6 ( see § 3.5.2) ( 5 files)

<p>5. <i>Send then a STEP command to the OPS subsystem :</i></p> <p><i>Enter STEP command in the PGF simulator window :</i></p> <p><b>STEP 1 input/IASI_1_wo_001bis</b></p> <p><i>Wait for the processing end.</i></p> <p>6. <i>Copy the processing data, needed to process the second granule, in the input directory (using symbolic links)</i></p>	<p>Verify that the STEP command is acknowledged.</p> <p>at the end of the processing from the directory \$WORK_DIR/OPS/data/TV/GRAN_OPS_N_07/output with the command :</p> <p><b>nedit CMD*</b></p> <p>Verify that all the input data are available with the commands :</p> <p><b>ls \$WORK_DIR/input</b></p> <p><b>ls \$WORK_DIR/input/product_model</b></p> <p><b>ls \$WORK_DIR/input/unproc_data</b></p>	
---	---	--

working order : IASI\_1\_wo\_002bis

IASI L0 data : J6 ( see § 3.5.2) ( second granule)

Product model : J6( see § 3.5.2) ( 5 files)

<p>7. <i>Send then a STEP command to the OPS subsystem :</i></p> <p><i>Enter STEP command in the PGF simulator window :</i></p> <p><b>STEP 2 input/IASI_1_wo_002bis</b></p> <p><i>Wait for the processing end.</i></p>	<p>Verify that the STEP command is acknowledged.</p> <p>at the end of the processing from the directory \$WORK_DIR/OPS/data/TV/GRAN_OPS_N_07/output with the command :</p> <p><b>nedit CMD*</b></p> <p>Verify that :</p> <ul style="list-style-type: none"> <li>log events trace the reinitialisation of the static configuration ,(other configuration file) due to a PTSI modification</li> </ul> <p>at the end of the processing from the directory \$WORK_DIR/OPS/data/TV/GRAN_OPS_N_07/output with the command :</p> <p><b>nedit TRACE*</b></p> <p>look for</p> <p>"CONFIGURATION FILE LOADING"</p> <p>and</p> <p>"PTSI HAS CHANGED"</p>	<p>[E_FIC5]</p> <p>[E_FIC6]</p>
--	---	---------------------------------

## 5.2.21.GRAN-OPS-N-08

Test Sheet	GRAN-OPS-N-08	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 processing in case of a modification of the instrument mode (calibration mode)</p> <p>Used data set : J10</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p> <p>If necessary uncompress recursively input data (directory \$WORK_DIR/../../tmp)</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test specific directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_N_08</pre> <p>then execute the test shell :</p> <pre>./lancetestbis.sh</pre> <p><b><i>This shell execute automatically the following steps (in italic style)</i></b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>	<p>Verify that <b>input</b>, <b>to_pgf</b> and <b>tmp</b> directories are purged by checking the dates.</p> <pre>ls -l \$WORK_ROOT_DIR/tmp</pre> <pre>ls -l \$WORK_ROOT_DIR/to_pgf</pre>	
<p>3. <i>Launch the OPS in standalone mode:</i></p> <pre>../../cmd/StartStandAlone.sh 1</pre> <p><b><i>\$WORK_DIR N</i></b></p>		
<p>4. <i>Copy at first, the input auxiliary data, needed to process all granules in the input directory (using symbolic links)</i></p>	<p>Verify that all the input data are available with the command :</p> <pre>ls \$WORK_DIR/input/aux_data</pre>	

AVHRR product : J10 ( see § 3.5.2)

Other and Stable Parameters , Spectral Database : J1 ( see § 3.5.2)

IASI Context file : J1 ( see § 3.5.2)

Auxiliary data : J1 ( see § 3.5.2)

5. *Copy the processing data, needed to process the granules, in the input directory (using symbolic links)*

Verify that all the input data are available with the commands :

**ls \$WORK\_DIR/input**

**ls \$WORK\_DIR/input/product\_model**

**ls \$WORK\_DIR/input/unproc\_data**

working order : IASI\_1\_wo\_001bis to IASI\_1\_wo\_008bis

IASI L0 data : J10 ( see § 3.5.2) (8 granules)

Product model : J10 ( see § 3.5.2) ( 5 files )

6. *Send then a STEP command for each granule to the OPS subsystem :*

*Enter STEP command in the PGF simulator window :*

**STEP 1 input/IASI\_1\_wo\_001bis ...**

*Wait for the processing end.*

Verify that the STEP commands are acknowledged.

- during the processing from the directory \$WORK\_DIR/tmp with the command :

**tail -f CMD\***

- at the end of the processing from the directory \$WORK\_DIR/OPS/data/TV/GRAN\_OPS\_N\_08/output with the command :

**nedit CMD\***

look for ACK STEP 1 0

ACK STEP 2 0

ACK STEP 3 0

ACK STEP 4 0

ACK STEP 5 0

ACK STEP 6 0

ACK STEP 7 0

ACK STEP 8 0

Verify that the processing chain stops correctly

**nedit CMD\***

look for ACK START 1 1

[\[PPS-3.6-030\]](#)

[\[PPF-0271\]](#)

<p>7. Verify the format of the product files</p> <p><i>cd output</i></p> <p><b>TraceProduit “product file name”</b></p> <p><b>GRANx</b></p>	<p>Each file is dumped (one of each type at least) with a validation tool (TraceProduit) to verify that the output products are in accordance with the GPFS format (number of IPR, order of the records, their size and offset).</p> <p>file ENG from the 1<sup>st</sup> granule</p> <p>file VER from the 1<sup>st</sup> granule</p> <p>file N1A from the 1<sup>st</sup> granule</p> <p>file N1B from the 1<sup>st</sup> granule</p> <p>file N1C from the 1<sup>st</sup> granule</p> <p><b>cd \$WORK_DIR/OPS/data/TV</b></p> <p><b>cd GRAN_OPS_N_08</b></p> <p><b>cd output/GRANx</b></p> <p><b>nedit structure.prod</b></p> <p><b>cd \$WORK_DIR/OPS/data/TV</b></p> <p><b>cd GRAN_OPS_N_08</b></p> <p><b>diff -s references/GRANx/structure.prod</b></p> <p><b>output/GRANx/structure.prod</b></p> <p>no difference must appear</p>	
<p>8. Analyse the content of the product (manual) for granule number two.</p> <p><b>cd output</b></p> <p><b>TraceProduit “product file name ”</b></p> <p><b>GRANx AUTO</b></p> <p><b>cd output</b></p> <p><b>TraceContexte « contex file name »</b></p> <p><b>contexte.txt</b></p>	<p>Verify that :</p> <ul style="list-style-type: none"> <li>The instrument mode is set to external calibration (using the output of TraceProduit of engineering product number 2)</li> </ul> <p><b>cd GRAN_ENG_2</b></p> <p><b>nedit MDR20.tra</b></p> <p>(search word ENG_INSTRUMENT_MODE that must be set to 162)</p> <ul style="list-style-type: none"> <li>The dedicated degraded case (CG3) is on (using the LOG messages),</li> </ul> <p><b>nedit LOG*</b></p> <p>( find DEGRADED CASE NUMBER 3)</p>	<p><a href="#">[E_MOD1]</a></p> <p><a href="#">[PPF-0630pt2]</a></p> <p><a href="#">[PPS-3.2-160]</a></p>

## 5.2.22.GRAN-OPS-N-09

Test Sheet	GRAN-OPS-N-09	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the initialisation stage of the IASI L1 processing and to verify the use of the platform mode (from OSV file) and reprocessing mode.</p> <p>Used data set : J9</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test specific directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/GRAN-OPS-N-09</pre> <p>then execute the test shell :</p> <pre>./lancetestbis.sh</pre> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <p>Connected as iasi_1 . You must go in the test specific directory :</p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>	Verify that : input, to_pgf and tmp directories are purged.	
<p>3. Launch the OPS :</p> <pre>../././cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetestbis.sh):</i></p>	The OPS is ready to process input data in nominal mode.	

4. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i>	Verify with the <b>ls \$WORK_DIR/input/aux_data</b> commands that all the input data are available :	
<u>AVHRR product</u> : J1 ( see § 3.5.2) <u>Other and Stable Parameters , Spectral Database</u> : J1 <u>IASI Context file</u> : J1 ( see § 3.5.2) <u>Auxiliary data</u> : J9 ( see § 3.5.2)		
5. <i>Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</i>	Verify with the <b>ls \$WORK_DIR/input</b> <b>ls \$WORK_DIR/input/product_model</b> <b>ls \$WORK_DIR/input/unproc_data</b> commands that all the input data are available :	
<u>working order</u> : IASI_1_wo_001 to IASI_1_wo_004 <u>IASI L0 data</u> : J1 ( see § 3.5.2) ( first four granule) <u>Product model</u> : J1R) ( 5 files)		
6. <i>Send then a STEP command to the OPS subsystem :</i>  <i>Enter STEP command in the PGF simulator window for each granule :</i>  <b>STEP 1 input/IASI_1_wo_001</b> .....  <i>Wait for the processing end.</i>  <i>Wait for the processing end.</i>	Verify that the STEP command is acknowledged.  Verify that in each product : <ul style="list-style-type: none"> <li>the processing mode is set to reprocessing</li> <li>the platform mode is set to “in plane maneuver”</li> </ul>	<a href="#">[E_MOD1]</a> <a href="#">[PPS-3.2-160]</a> <a href="#">[PPS-3.2-170]</a> <a href="#">PPS-3.2-420pt2 et 3]</a> <a href="#">[E_DEG6]</a> <a href="#">[PPS-3.6-030]</a>
8. Verify the content- of the product files (manual)  <b>cd output</b>  <b>TraceProduit “ product file name ”</b> <b>GRANx</b>  <b>cd GRANx</b>	The first ENG Product file is dumped with. A validation tool to verify that the output product are in accordance with the GPFS format. (for exemple , the MPHR and the first MDR )  <b>more MPHR.tra</b> ( the key word MPHR_PROCESSING_MODE must be set to R)  <b>more MDR1.tra</b> (the Key word “ENG_REPROCESSING_MODE “must be set to 1)  (the Key word “ENG_MANOEUVRE “must be set to 1)	



## 5.2.23.GRAN-OPS-D-10

Test Sheet	GRAN-OPS-D-10	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 chain with degraded input data that can be therefore processed.</p> <p>Used data set : from J2.1 to J2.13</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test specific directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_D_10 en</pre> <p>execute the test shell :</p> <pre>./lancetestbis.sh</pre> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <p>Connected as iasi_1 . You must go in the test specific directory :</p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>	Verify that : input, to_pgf and tmp directories are purged.	
<p>3. Launch the OPS :</p> <pre>./.././../cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetestbis.sh):</i></p>	The OPS is ready to process input data in nominal mode.	

<p>4. <i>Copy at first, the input auxiliary data, needed to process all granules in the input directory (using symbolic links)</i></p>	<p>Verify that all the input auxilliary data are available with the command :</p> <p><b>ls \$WORK_DIR/input/aux_data</b></p>	
<p><u>AVHRR product</u> : J1 J2.11 J2.12 ( see § 3.5.2)</p> <p><u>Other and Stable Parameters , Spectral Database</u> : J1 ( see § 3.5.2)</p> <p><u>IASI Context file</u> : J1 (see § 3.5.2) this file is copied from the delivery</p> <p><u>Auxiliary data</u> : J1 ( see § 3.5.2)</p>		
<p>5. <i>Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</i></p>	<p>Verify that all the input data are available with the commands :</p> <p><b>ls \$WORK_DIR/input</b></p> <p><b>ls \$WORK_DIR/input/product_model</b></p> <p><b>ls \$WORK_DIR/input/unproc_data</b></p>	
<p><u>working order</u> : IASI_1_wo_001bis</p> <p><u>IASI L0 data</u> : J2.1 ( see § 3.5.2)</p> <p><u>Product model</u> : J1 ( see § 3.5.2)</p>		

<p>6. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP 1 input/IASI_1_wo_001bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is acknowledged.</p> <p>– during the processing from the directory \$WORK_DIR/tmp with the command :</p> <p><b>tail -f CMD*</b></p> <p>- at the end of the processing from the directory \$WORK_DIR/OPS/data/TV/GRAN_OPS_D_10/output with the command :</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 1 0</p> <p>Verify that the processing chain stops correctly</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 1 1</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> <p>A Log event is send to mention the detection of a degraded case)</p> <p><b>cd output</b></p> <p><b>nedit LOGxxx*</b></p> <p>look for the folloowing key words</p> <p>“DEGRADED CASE NUMBER 4 HAS BEEN RAISED” and</p> <p>“DEGRADED CASE NUMBER 1 HAS BEEN RAISED”</p> <p>"DEGRADED CASE NUMBER 3 HAS BEEN RAISED" for SN #7</p>	<p><a href="#">[PPS-3.2-090]</a></p> <p><a href="#">[PPF-0641]</a></p>
<p>7. Verify the content of the product files (manual )</p> <p><b>cd output/J2.1</b></p> <p><b>TraceProduit “ENG product file name” GRANx ( MDR number 3)</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with a validation tool (TraceProduit).</p> <p>This is done in order to verify that the output product has a specific flag for the degraded pixel ( CG1 degraded case) for the dedicated LN,SN,PN</p> <p><b>nedit *.tra</b></p> <p>Find key-word “GOPSFLTBMANDMISS” and “GOPSFLAPIXMISS” (values must be set to 1 according to V1 Test Data set).</p>	<p><a href="#">[E_DEG0]</a></p> <p><a href="#">[E_DEG7]</a></p>

<p>8. Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</p>	<p>Verify that all the input data are available with the commands :</p> <p><b>ls \$WORK_DIR/input</b></p> <p><b>ls \$WORK_DIR/input/product_model</b></p> <p><b>ls \$WORK_DIR/input/unproc_data</b></p>	
<p><u>working order</u> : IASI_1_wo_002bis</p> <p><u>IASI L0 data</u> : J2.1bis ( see § 3.5.2)</p> <p><u>Product model</u> : J1bis ( see § 3.5.2)</p>		
<p>9. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP 2 input/IASI_1_wo_002bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is acknowledged.</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 2 0</p> <p>Verify that chain stops correctly</p> <p>look for ACK STEP 2 1</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file : A Log event is send to mention the detection of a degraded case).</li> </ul> <p><b>cd output</b></p> <p><b>more LOGxxx*</b></p> <p>look for the following key words</p> <p>“DEGRADED CASE NUMBER 1 HAS BEEN RAISED”</p>	<p><a href="#">[PPS-3.2-090]</a></p> <p><a href="#">[PPF-0641]</a></p>
<p>10. Verify the content- of the product files (manual)</p> <p><b>cd output/J2.1bis</b></p> <p><b>TraceProduit “ENG product file name” GRANx ( MDR number 3)</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with a validation tool (TraceProduit).</p> <p>This is done in order to verify that the output product has a specific flag for the lost pixel ( CG1 degraded case) for the dedicated LN,SN,PN</p> <p><b>nedit *.tra</b></p> <p>Find key-words “GOPSFLAPIXMISS” and “ISRFEMOFF” ( values must be set to 1 according to V1 Test Data set.</p>	<p><a href="#">[E_DEG0]</a></p> <p><a href="#">[E_DEG7]</a></p>
<p>11. Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</p>		

working order : IASI\_1\_wo\_003bis

IASI L0 data : J2.2 ( see § 3.5.2)

Product model : J1 ( see § 3.5.2)

<p>12. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP3 input/IASI_1_wo_003bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is acknowledged.</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 3 0</p> <p>Verify that the processing chain stops correctly</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 3 1</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> <p>A Log event is send to mention the detection of a degraded case)</p> <p><b>cd output</b></p> <p><b>nedit LOGxxx*</b></p> <p>look for the following key words</p> <p>“DEGRADED CASE NUMBER 13 HAS BEEN RAISED” and</p> <p>“DEGRADED CASE NUMBER 1 HAS BEEN RAISED”</p> <p>“DEGRADED CASE NUMBER 3 HAS BEEN RAISED”</p>	<p><a href="#">[PPS-3.2-090]</a></p> <p><a href="#">[PPF-0641]</a></p>
<p>13. Verify the content- of the product (manual)files</p> <p><b>cd output/J2.2</b></p> <p><b>TraceProduit “ENG product file name” GRANx ( MDR number 3)</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with a validation tool (TraceProduit).</p> <p>This is done in order to verifiye that the output product has a specific flag for the lost pixel ( CG3 degraded case) for the dedicated LN,SN ( ISRFEM processing is inactivate)</p> <p><b>nedit *.tra</b></p> <p>Find key-word “GOPSFLAGPACKETPXMISS” and “GOPSFLTLISRFEMOFF” ( values must be set to 1 according to V1 Test Data set).</p>	<p><a href="#">[E_DEG7]</a></p>
<p>14. Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</p>		

working order : IASI\_1\_wo\_004bis

IASI L0 data : J2.3 ( see § 3.5.2)

Product model : J1 ( see § 3.5.2)

<p>15. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP 4 input/IASI_1_wo_004bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is acknowledged.</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 4 0</p> <p>Verify that the processing chain stops correctly</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 4 1</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> <p>No Log event is send to mention the detection of a degraded case (the data gap can be showed in the product stucture)</p>	
<p>16. Verify the content- of the product files (manual).</p> <p>cd output/J2.3</p> <p><b>TraceProduit “ENG product file name” GRANx ( MDR number 3)</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with a validation tool (TraceProduit).</p> <p>Thi is done in order to verify that the output product is coherent with the data gap:</p> <ul style="list-style-type: none"> <li>Nominal processing</li> <li>Data gap in the IASI L1 product</li> </ul> <p><b>nedit structure.prod</b></p> <p>(record 11 and 12 must be dummy (subclass=1))</p>	
<p>17. Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</p>		
<p><u>working order</u> : IASI_1_wo_005bis</p> <p><u>IASI L0 data</u> : J2.4 ( see § 3.5.2)</p> <p><u>Product model</u> : J1 ( see § 3.5.2)</p>		

<p>18. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP 5 input/IASI_1_wo_005bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is acknowledged.</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 5 0</p> <p>Verify that the processing chain stops correctly</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 5 1</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> <p>A Log event is send to mention the detection of a degraded case)</p> <p><b>cd output</b></p> <p><b>nedit LOGxxx*</b></p> <p>look for the following key words</p> <p>“DEGRADED CASE NUMBER 4 HAS BEEN RAISED”</p>	<p><a href="#">[PPS-3.2-090]</a></p> <p><a href="#">[PPF-0641]</a></p>
<p>19. Verify the content- of the product files (manual)</p> <p><b>Cd output/J2.4</b></p> <p><b>TraceProduit “ENG product file name” GRANx ( MDR number 3)</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with a validation tool (TraceProduit).</p> <p>This is done in order to verify that the output product has a specific flag for the pixel containing underflow/overflow (this is done thanks to 20_DOC algorithm)</p> <p><b>nedit *.tra</b></p> <p>Find key-word “GOPSFLT BANDMISS” and “GOPSFLTISRFE MOFF” ( values must be set to 1 according to V1 Test Data set.</p> <p>Find Kew word “GDOCNB UNDERFLOW” and “GDOCNB OVERFLOW”( values must be set to 1 according to V1 Test Data set)</p>	<p><a href="#">[E_DEG7]</a></p>
<p>20. Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</p>		
<p><u>working order</u> : IASI_1_wo_006bis</p> <p><u>IASI L0 data</u> : J2.5 ( see § 3.5.2)</p> <p><u>Product model</u> : J1 ( see § 3.5.2)</p>		

<p>21. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP 6 input/IASI_1_wo_006bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that each STEP command is acknowledged.</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 6 0</p> <p>Verify that the processing chain stops correctly</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 6 1</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> <p>A Log event is send to mention the detection of a degraded case)</p> <p><b>cd output</b></p> <p><b>nedit LOGxxx*</b></p> <p>look for the following key words “DEGRADED CASE NUMBER 12 HAS BEEN RAISED” and “DEGRADED CASE NUMBER 5 HAS BEEN RAISED”</p> <p>Degraded case number 5 must be permanent</p>	<p><a href="#">[PPS-3.2-090]</a></p> <p><a href="#">[PPF-0641]</a></p>
<p>22. Verify the content- of the product files (manual)</p> <p><b>Cd output/J2.5</b></p> <p><b>TraceProduit “ENG product file name” GRANx ( MDR number 3)</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with a validation tool (TraceProduit).</p> <p>This is done in order to verify that the output product has a specific flag for the lost pixel ( CG12 degraded cases) for the dedicated LN,)</p> <p><b>nedit *.tra</b></p> <p>Find key-word “GOPSFLAGPACKETAPMISS” and “GOPSFLTLBBTMISS” ( values must be set to 1 according to V1 Test Data set.</p>	<p><a href="#">[E_DEG1]</a></p> <p><a href="#">[E_DEG7]</a></p>
<p>23. Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</p>		
<p><u>working order</u> : IASI_1_wo_007bis</p> <p><u>IASI L0 data</u> : J2.6 ( see § 3.5.2)</p> <p><u>Product model</u> : J1 ( see § 3.5.2)</p>		



<p>24. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP 7 input/IASI_1_wo_007bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is acknowledged.</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 7 0</p> <p>Verify that the processing chain stops correctly</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 7 1</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> <p>A Log event is send to mention the detection of a degraded case)</p> <p><b>cd output</b></p> <p><b>nedit LOGxxx*</b></p> <p>look for the following key words</p> <p>“DEGRADED CASE NUMBER 14 HAS BEEN RAISED” and</p> <p>“DEGRADED CASE NUMBER 7 HAS BEEN RAISED”</p> <p>Degraded case number 7 must be permanent</p>	<p><a href="#">[PPS-3.2-090]</a></p> <p><a href="#">[PPF-0641]</a></p>
<p>25. Verify the content- of the product files (manual).</p> <p><b>cd output/J2.6</b></p> <p><b>TraceProduit “ENG product file name” GRANx ( MDR number 3, 5 and 6)</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with a validation tool (Traceproduit).</p> <p>This is done in order to verify that the output product has a specific flag for the lost image ( CG14 and CG7 degraded cases) for the dedicated LN,)</p> <p><b>nedit *.tra</b></p> <p>Find key-word “GOPSFLTIMGBBMISS” ( values must be set to 1 according to V1 Test Data set.</p> <p>For the field "GOPSDATIMGBBMISS", the date of MDR #5 must be greater than the date of MDR #3 and the date of MDR # 6 must be the same as the date of MDR # 5.</p>	<p><a href="#">[E_DEG7]</a></p>
<p>26. Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</p>		
<p><u>working order</u> : IASI_1_wo_008bis</p> <p><u>IASI L0 data</u> : J2.7 ( see § 3.5.2)</p> <p><u>Product model</u> : J1 ( see § 3.5.2)</p>		

<p>27. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP87 input/IASI_1_wo_008bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is acknowledged.</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 8 0</p> <p>Verify that the processing chain stops correctly</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 8 1</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> <p>A Log event is send to mention the detection of a degraded case)</p> <p><b>cd output</b></p> <p><b>nedit LOGxxx*</b></p> <p>look for the following key words</p> <p>“DEGRADED CASE NUMBER 14 HAS BEEN RAISED” and</p> <p>“DEGRADED CASE NUMBER 8 HAS BEEN RAISED”</p>	<p><a href="#">[PPS-3.2-090]</a></p> <p><a href="#">[PPF-0641]</a></p>
<p>28. Verify the content- of the product files (manual)</p> <p><b>cd output/J2.7</b></p> <p><b>TraceProduit “ENG product file name” GRANx ( MDR number 3, 4 and 5)</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with a validation tool (TraceProduit).</p> <p>This is done in order to verify that the output product has a specific flag for the lost image ( CG14 and CG8 degraded cases) for the dedicated LN,)</p> <p><b>nedit *.tra</b></p> <p>Find key-word “GOPSFLTIMGCSMISS ( values must be set to 1 according to V1 Test Data set.</p> <p>Verify GOPSDATIMGCSMISS.</p>	<p><a href="#">[E_DEG2]</a></p> <p><a href="#">[E_DEG7]</a></p>
<p>29. Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</p>		
<p><u>working order</u> : IASI_1_wo_009bis</p> <p><u>IASI L0 data</u> : J2.8 ( see § 3.5.2)</p> <p><u>Product model</u> : J1 ( see § 3.5.2)</p>		

<p>30. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP 9 input/IASI_1_wo_009bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is acknowledged.</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 9 0</p> <p>Verify that the processing chain stops correctly</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 9 1</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file : A Log event is send to mention the detection of a degraded case)</li> </ul> <p><b>cd output</b> <b>nedit LOGxxx*</b> "DEGRADED CASE NUMBER 14 HAS BEEN RAISED" and "DEGRADED CASE NUMBER 6 HAS BEEN RAISED" and "DEGRADED CASE NUMBER 3 HAS BEEN RAISED"</p>	<p><a href="#">[PPS-3.2-090]</a> <a href="#">[PPF-0641]</a></p>
<p>31. Verify the content- of the product files (manual)</p> <p><b>cd output/J2.8</b></p> <p><b>TraceProduit "ENG product file name" GRANx ( MDR number 3)</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with a validation tool (TraceProduit).</p> <p>This is done in order to verify that the output product has a specific flag for the lost image ( CG14 and CG6 degraded cases) for the dedicated LN,).</p> <p><b>nedit *.tra</b></p> <ul style="list-style-type: none"> <li>Find key-word "GOPSFLTIMGWMISS", "GOPSFLAPACKETIPMISS" and "GOPSFLTISRFEWMISS" ( values must be set to 1 according to V1 Test Data set.</li> </ul>	<p><a href="#">[E_DEG2]</a> <a href="#">[E_DEG7]</a></p>
<p>32. Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</p>		
<p><u>working order</u> : IASI_1_wo_0010bis</p> <p><u>IASI L0 data and AVHRR data</u>: J2.10 ( see § 3.5.2)</p> <p><u>Product model</u> : J1 ( see § 3.5.2)</p>		

<p>33. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP 10 input/IASI_1_wo_010bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is acknowledged.</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 10 0</p> <p>Verify that the processing chain stops correctly</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 10 1</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> <p>A Log event is send to mention the detection of a degraded case)</p> <p><b>cd output</b></p> <p><b>nedit LOGxxx*</b></p> <p>“DEGRADED CASE NUMBER 9 HAS BEEN RAISED”</p>	<p><a href="#">[PPS-3.2-090]</a></p> <p><a href="#">[PPF-0641]</a></p>
<p>34. Verify the content- of the product files (manual)</p> <p><b>Cd outpur/J2.10</b></p> <p><b>TraceProduit “ENG product file name” GRANx ( MDR number 2)</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with a validation tool (TraceProduit).</p> <p>This is done in order to verify that the output product has a specific flag for the missing AVHRR data ( CG9 degraded cases) for the dedicated LN</p> <p>The granule is processed.but the AVHRR offset is not computed.</p> <p><b>nedit *.tra</b></p> <p>Find key-word and</p> <p>“GOPSFLTRADAVHRRMISS” ( values must be set to 1 according to V1 Test Data set.</p>	<p><a href="#">[E_DEG2]</a></p> <p><a href="#">[E_DEG7]</a></p>
<p>35. Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</p>		
<p><u>working order</u> : IASI_1_wo_0011bis</p> <p><u>IASI L0 data and AVHRR data</u>: : J2.11 ( see § 3.5.2)</p> <p><u>Product model</u> : J1 ( see § 3.5.2)</p>		

<p>36. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP 11 input/IASI_1_wo_011bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is acknowledged.</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 11 0</p> <p>Verify that with the last granule the processing chain stops</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 11 1</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file : A Log event is send to mention the detection of a degraded case)</li> </ul> <p><b>cd output</b> <b>nedit LOGxxx*</b> "DEGRADED CASE NUMBER 15 HAS BEEN RAISED"</p>	<p><a href="#">[PPS-3.2-090]</a> <a href="#">[PPF-0641]</a></p>
<p>37. Verify the content- of the product files (manual)</p> <p><b>cd output/J2.11</b></p> <p><b>TraceProduit "ENG product file name" GRANx ( MDR number 14)</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with a validation tool (TraceProduit).</p> <p>This is done in order to verify that the output product has a specific flag for the incorrect AVHRR geolocalisation data ( CG15 degraded cases) for the dedicated LN</p> <p>The granule is processed.but the AVHRR offset is not computed.</p> <p><b>nedit *.tra</b></p> <p>Find key-word "GOPSFLTRADAVHRRMISS" ( values must be set to 1 according to V1 Test Data set.</p>	<p><a href="#">[E_DEG7]</a></p>
<p>38. Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</p>		
<p><u>working order</u> : IASI_1_wo_0012bis</p> <p><u>IASI L0 data</u> : J2.12 ( see § 3.5.2)</p> <p><u>Product model</u> : J1 ( see § 3.5.2)</p>		

<p>39. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP 12 input/IASI_1_wo_012bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that each STEP command is acknowledged.</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 12 0</p> <p>Verify that the processing chain stops correctly</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 12 1</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> <p>A Log event is send to mention the detection of a degraded case)</p> <p><b>cd output</b></p> <p><b>nedit LOGxxx*</b></p> <p>“DEGRADED CASE NUMBER 11 HAS BEEN RAISED”</p>	<p><a href="#">[PPS-3.2-090]</a></p> <p><a href="#">[PPF-0641]</a></p>
<p>40. Verify the content- of the product files (manual)</p> <p><b>Cd output/J2.12</b></p> <p><b>TraceProduit “ENG product file name” GRANx ( MDR number 4 and 7)</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with a validation tool (TraceProduit).</p> <p>This is done in order to verify that the output product has a specific flag for the lost pixel ( CG11 degraded case) for the dedicated LN,) the verification product is filled with dummy data (0)</p> <p><b>nedit *.tra</b></p> <p>Find key-word and</p> <p>“GOPSFLAGPACKETVPMISS” ( values must be set to 1 according to V1 Test Data set. )</p>	<p><a href="#">[E_DEG7]</a></p>
<p>41. Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</p>		
<p><u>Working order</u> : IASI_1_wo_0013bis</p> <p><u>IASI L0 data</u> : J2.13 ( see § 3.5.2)</p> <p><u>Product model</u> : J2.13 ( see § 3.5.2)</p>		

<p>42. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP 13 input/IASI_1_wo_013bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that each STEP command is acknowledged.</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 13 0</p> <p>Verify that the processing chain stops correctly</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 13 0</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> <p>Several Log event is send to mention the detection of degraded cases</p> <p><b>cd output</b></p> <p><b>nedit LOGxxx*</b></p> <p>“DEGRADED CASE NUMBER 5 HAS BEEN RAISED</p> <p>“DEGRADED CASE NUMBER 3 HAS BEEN RAISED</p> <p>“DEGRADED CASE NUMBER 6 HAS BEEN RAISED</p> <p>“DEGRADED CASE NUMBER 8 HAS BEEN RAISED</p> <p>“DEGRADED CASE NUMBER 7 HAS BEEN RAISED</p>	<p><a href="#">[PPS-3.2-090]</a></p> <p><a href="#">[PPF-0641]</a></p> <p><a href="#">[E_FLA1]</a></p>
<p>43. Verify the content- of the product files (manual)</p> <p>Cd output/J2.13</p> <p><b>TraceProduit “ENG product file name” GRANx ( MDR number 4, 10, 17)</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with a validation tool (TraceProduit).</p> <p>This is done in order to verify that the output product has specific flags for degraded data ( CGxx degraded case) for the dedicated LN,)</p> <p><b>nedit *.tra</b></p> <p>Find key-word “GOPSIMGEWMISS”and “GOPSFLTBBTMISS” (values must be set to 1 according to V1 Test Data set. ).</p> <p>Key word “GOPSIMGEWMISS” must be set to 0 for MDR #4</p>	<p><a href="#">[E_DEG7]</a></p>

## 5.2.24.GRAN-OPS-D-11

Test Sheet	GRAN-OPS-D-11	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 chain with degraded input leading to a permanent degraded case.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Launch the OPS :</p> <p>Execute the first 3 steps of RUN-OPS-N-01 test.</p>	The OPS is ready to process input data in nominal mode.	
<p>2. Connected as iasi_1 then go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_D_11</code></p> <p>then execute the test shell :</p> <p><b><code>./lancetestbis.sh</code></b></p> <p>This shell executes automaticaly the following steps, in italic style</p>		



<p>3. <i>Copy(using symbolic links) all data needed to process both granules in the directories</i>  <i>\$WORK_DIR/input/aux_data</i>  <i>\$WORK_DIR/input/unproc_data</i>  <i>\$WORK_DIR/input/product_model</i></p>	<p>Data are copied in  <b>\$WORK_DIR/input/aux_data</b>  <b>\$WORK_DIR/input/unproc_data</b>  <b>\$WORK_DIR/input/product_model</b></p> <p>A screen message confirms the operation :  <i>mise en place des données auxiliaires</i>  <i>Copy SUCCESS</i>  <i>END</i>  <i>mise en place des product model</i>  <i>Copy SUCCESS</i>  <i>END</i><i>mise en place des données d'entrée</i>  <i>Copy SUCCESS</i>  <i>END</i></p>	
<p><u>AVHRR product</u> : J1 ( see § 3.5.2)  <u>Other and Stable Parameters , Spectral Database</u> : TV_THA_CON_1 ( see § 3.5.2)  <u>IASI Context file</u> : J1 (see § 3.5.2) this file is copied from the delivery  <u>Auxiliary data</u> : J1 ( see § 3.5.2)</p>		
<p><u>working order</u> : IASI_1_wo_001  <u>IASI L0 data</u> : J2.13 ( see § 3.5.2)  <u>Product model</u> : J2.13 ( see § 3.5.2)</p>		

<p>4. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is acknowledged.</p> <p>ACK STEP 1 0</p> <p>Verify that WorkingOrder 1 as begun :</p> <p>ACK START 1 0</p> <p>Verify the process terminates correctly =&gt; screen message ACK START 1 1</p> <p>Verify after step 6 that :</p> <ul style="list-style-type: none"> <li>A logEvent is generated for the first granule due to degraded case detection (CG5, CG3, CG6 in the same order) :</li> </ul> <p>DEGRADED CASE NUMBER 5 HAS BEEN RAISED</p> <p>DEGRADED CASE NUMBER 3 HAS BEEN RAISED</p> <p>DEGRADED CASE NUMBER 6 HAS BEEN RAISED</p> <ul style="list-style-type: none"> <li>CG3 and CG6 become permanent</li> </ul> <p>DEGRADED CASE NUMBER 3 PERMANENT</p> <p>DEGRADED CASE NUMBER 6 PERMANENT</p> <p>Verify that the predicted files are produced :</p> <p>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated:</p>	<p>[[PPS-3.2-100]</p> <p>[PPF-0120]</p> <p><a href="#">[E_DEG7]</a></p> <p><a href="#">[PPS-3.2-090]</a></p>
<p>5. Verify the content- of the product files</p>	<p>Each file is dumped with. a validation tool :</p> <p><i>cd \$WORK_DIR/output</i></p> <p><i>TraceProduit ProdName dir</i></p> <p>This done in order to verify that the output product has a specific flag for losts pixels ( CG1 degraded case) for the dedicated LN,SN,PN</p> <p>Vérify CG3 is raised for LN3 and CG4 is raised for LN3,4,5 and becomes permanent</p>	

## 5.2.25.GRAN-OPS-D-12

Test Sheet	GRAN-OPS-D-12	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the degraded case in case of a missing auxiliary file (OSV file)</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Launch the OPS :</p> <p>Execute the first 3 steps of RUN-OPS-N-01 test.</p>	The OPS is ready to process input data in nominal mode.	
<p>2. Connected as iasi_1 then go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_D_12</code></p> <p>then execute the test shell :</p> <p><code>./lancetestbis.sh</code></p> <p>This shell executes automaticaly the following steps, in italic style</p>		

<p>3. <i>Copy(using symbolic links) all data needed to process both granules in the directories</i>  <i>\$WORK_DIR/input/aux_data</i>  <i>\$WORK_DIR/input/unproc_data</i>  <i>\$WORK_DIR/input/product_model</i></p>	<p>Data are copied in  <b>\$WORK_DIR/input/aux_data</b>  <b>\$WORK_DIR/input/unproc_data</b>  <b>\$WORK_DIR/input/product_model</b></p> <p>A screen message confirms the operation :  <i>mise en place des données auxiliaires</i>  <i>Copy SUCCESS</i>  <i>END</i></p> <p><i>mise en place des product model</i>  <i>Copy SUCCESS</i>  <i>END</i></p> <p><i>mise en place des données d'entrée</i>  <i>Copy SUCCESS</i>  <i>END</i></p>	
<p><u>AVHRR product</u> : J1 ( see § 3.5.2)</p> <p><u>Other and Stable Parameters , Spectral Database</u> : J1 ( see § 3.5.2)</p> <p><u>IASI Context file</u> : J1 ( see § 3.5.2)</p> <p><u>Auxiliary data</u> : J1 ( see § 3.5.2) sans fichier OSV</p>		
<p><u>working order</u> : IASI_1_wo_001</p> <p><u>IASI L0 data</u> : J1 ( see § 3.5.2) ( first granule)</p> <p><u>Product model</u> : J12 (see § 3.5.2) (5 files)</p>		

<p>4. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window :</p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is acknowledged.</p> <p>ACK STEP 1 0</p> <p>Verify that WorkingOrder 1 as begun :</p> <p>ACK START 1 0</p> <p>Verify the process terminates correctly =&gt; screen message ACK START 1 1</p> <p>Verify after step 1 that :</p> <ul style="list-style-type: none"> <li>• A logEvent is send to indicate the absence of OSV file : DEFAULT INITIALISATION IS DONE FOR OSV DATA</li> <li>• log events trace the correct execution of the order and the L1 products generation</li> </ul> <p>Dump one of the generated product :</p> <p>cd \$WORK_DIR/output</p> <p>TraceProduit IASI_ENG_xx_xxxxxx dir</p> <p>(Choose MDR dumping mode)</p> <p>Verify that in each line and in the MPHR:</p> <ul style="list-style-type: none"> <li>• the processing mode is set to reprocessing</li> <li>• the MANOEUVRE flag mode is set to normal</li> </ul>	<p><a href="#">[PPF-0850pt2]</a></p>
--	--	--------------------------------------

## 5.2.26.GRAN-OPS-D-13

Test Sheet	GRAN-OPS-D-13	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate. the IASI L1 processing using aberrant data.</p> <p>Used data set : J15</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Launch the OPS in investigation mode:</p> <p>Execute the first 3 steps of RUN-OPS-N-04 test.</p>	<p>The OPS is ready to process input data in investigation mode.</p>	
<p>2. Connected as iasi_1 . You must go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/GRAN-OPS-D-13/ input</code></p> <p>Edit unproc_data file and change J1 to J15 for context file</p>		
<p>1. Execute the test shell :</p> <p><code>./lancetestbis.sh</code></p> <p><b>This shell execute automaticaly the following steps (<i>in italic style</i>)</b></p>		
<p>3. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i></p>	<p>Verify with the</p> <p><b>ls \$WORK_DIR/input/aux_data</b></p> <p>commands that all the input data are available :</p>	

<u>AVHRR product</u> : J1 ( see § 3.5.2) <u>Other and Stable Parameters , Spectral Database</u> : J1 ( see § 3.5.2) <u>IASI Context file</u> : J15 (see § 3.5.2) this file is copied from the delivery <u>Auxiliary data</u> : J1 ( see § 3.5.2)		
4. <i>Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</i>	Verify with the <b>ls \$WORK_DIR/input</b> <b>ls \$WORK_DIR/input/product_model</b> <b>ls \$WORK_DIR/input/unproc_data</b> commands that all the input data are available :	
<u>working order</u> : IASI_1_wo_001 <u>IASI L0 data</u> J1 ( see § 3.5.2) ( one granule) <u>Product model</u> : J1 ( see § 3.5.2) (all granule)		
5. <i>Send then a STEP command to the OPS subsystem :</i>  <i>Enter STEP command in the PGF simulator window for all the granules :</i>  <b>STEP 1 input/IASI_1_wo_001</b> .....  <i>Wait for the processing end.</i>	Verify the STEP command is acknowledged. • Verify the chain stops on error Verify no product is generated  Verify in the TRACE file that the Context file generated an error that led to stop the chain	<a href="#">[PPS-3.2-490]</a> [PPF-0420]
6. Replay all previous steps changing context file from J15 to J1 and Spectral Database from J1 to J15	• Verify the chain stops on error Verify no product is generated  Verify in the TRACE file that the Spectral Database file generated an error that led to stop the chain	
7. Replay all previous steps changing Spectral Database file from J15 to J1 and Stable Parameter file from J1 to J15	• Verify the chain stops on error Verify no product is generated  Verify in the TRACE file that Stable Parameter file generated an error that led to stop the chain	
9. Replay all previous steps changing Stable Parameter file from J15 to J1 and Other Parameter file from J1 to J15	• Verify the chain stops on error Verify no product is generated  Verify in the TRACE file that Other Parameter file generated an error that led to stop the chain	

## 5.2.27.GRAN-OPS-D-14

Test Sheet	GRAN-OPS-D-14	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate. the IASI L1 processing using dummy data</p> <p>Used data set : J1cing</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Launch the OPS :</p> <p>Execute the first 3 steps of RUN-OPS-N-01 test.</p>	The OPS is ready to process input data in nominal mode.	
<p>2. Connected as iasi_1 then go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_D_14</code></p> <p>then execute the test shell :</p> <p><b><code>./lancetestbis.sh</code></b></p> <p>This shell executes automaticaly the following steps, in italic style</p>		



<p>3. <i>Copy(using symbolic links) all data needed to process both granules in the directories</i>  <i>\$WORK_DIR/input/aux_data</i>  <i>\$WORK_DIR/input/unproc_data</i>  <i>\$WORK_DIR/input/product_model</i></p>	<p>Data are copied in  <b>\$WORK_DIR/input/aux_data</b>  <b>\$WORK_DIR/input/unproc_data</b>  <b>\$WORK_DIR/input/product_model</b></p> <p>A screen message confirms the operation :  <i>mise en place des données auxiliaires</i>  <i>Copy SUCCESS</i>  <i>END</i>  <i>mise en place des product model</i>  <i>Copy SUCCESS</i>  <i>END</i>  <i>mise en place des données d'entrée</i>  <i>Copy SUCCESS</i>  <i>END</i></p>	
<p><u>AVHRR product</u> : J1 ( see § 3.5.2)  <u>Other and Stable Parameters , Spectral Database</u> : J1 ( see § 3.5.2)  <u>IASI Context file</u> : J1 (see § 3.5.2) this file is copied from the delivery  <u>Auxiliary data</u> : J1 ( see § 3.5.2)</p>		
<p><u>working order</u> : IASI_1_wo_001bis  <u>IASI L0 data filled with non representative data (0 for ex):</u> J1cinq ( see § 3.5.2) ( one granule)  <u>Product model</u> : J1cinq ( see § 3.5.2) (one granule)</p>		
<p>4. <i>Send then a STEP command to the OPS subsystem :</i>   <i>Enter STEP command in the PGF simulator window :</i>   <i>STEP 1 input/IASI_1_wo_001bis</i>    <i>Wait for the processing end.</i></p>	<p>Verify that the STEP command is acknowledged.  <b>nedit CMD*</b>  look for ACK STEP 1 0</p> <p>Verify the process terminates correctly  <b>nedit CMD*</b>  look for ACK START 1 1</p> <p>Verify that :</p> <ul style="list-style-type: none"> <li>The predicted files are produced : IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> </ul>	<p><a href="#">[PPS-3.2-490]</a></p>

5. Verify the content of the product files	<p>Each file is dumped with a validation tool (TraceProduit ) to verify that the output product are in accordance with the GPFS format :</p> <p><b>cd \$WORK_DIR/OPS/data/TV</b></p> <p><b>cd GRAN_OPS_D_14/output</b></p> <p><b>TraceProduit &lt;PRODUCT_L1C&gt;</b></p> <p><b>GRAN_&lt;LABEL_L1C&gt;</b></p> <p>The data produced must be filled with dummy MDR</p> <p><b>cd GRAN_&lt;LABEL_L1C&gt;</b></p> <p><b>nedit structure prod</b></p> <p>verify that :</p> <p>MDR size = 21</p> <p>MDR subclass = 1</p>	
--	--	--

## 5.2.28.GRAN-OPS-D-15

Test Sheet	GRAN-OPS-D-15	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 chain with missing AVHRR data. Products can be therefore processed.</p> <p>Used data set : J2.9</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test specific directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_D_15</pre> <p>then execute the test shell :</p> <pre>.. /lancetestbis.sh</pre> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>	Verify that : input, to_pgf and tmp directories are purged.	
<p>3. <i>Launch the OPS in Stand alone mode.</i></p> <pre>../././cmd/StartStandAlone.sh 1 \$WORK_DIR N (this step is included in the shell lancetestbis.sh)</pre>	The OPS is ready to process input data in nominal mode:	
<p>4. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i></p>	Verify with the  <b>ls \$WORK_DIR/input/aux_data</b> commands that all the input data are available :	

<u>AVHRR product</u> : J1 ( see § 3.5.2) <u>Other and Stable Parameters , Spectral Database</u> : J1 ( see § 3.5.2) <u>IASI Context file</u> : J1 (see § 3.5.2) this file is copied from the delivery <u>Auxiliary data</u> : J1 ( see § 3.5.2)		
5. <i>Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</i>	:	
<u>working order</u> : IASI_1_wo_00x <u>IASI L0 data</u> : (7 granules) and AVHRR data : J2.9 ( see § 3.5.2) <u>Product model</u> : J2.9 ( see § 3.5.2) <u>(no AVHRR file)</u>		
6. Send then a STEP command to the OPS subsystem : <i>Enter STEP command in the PGF simulator window for all the granules :</i> <b>STEP 9 input/IASI_1_wo_009</b>  <i>Wait for the processing end.</i>	Verify that each STEP command is acknowledged. Verify that with the last granule the processing chain stops Verify that the predicted files are produced : <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file : A Log event is send to mention the detection of a degraded case (CG9)</li> </ul>	<a href="#">[PPS-3.2-090]</a>
7. Verify the content- of the product files (manual)  <b>cd output</b> <b>TraceProduit “ product file name ”</b> <b>GRANx AUTO</b>  8.	One product of type ENG is dumped with. a validation tool.  The granule is processed.but the AVHRR offset is not computed.	<a href="#">[E_DEG2]</a> <a href="#">[E_DEG7]</a> <a href="#">[PPF-0870]</a>

## 5.2.29.GRAN-OPS-N-16

Test Sheet	GRAN-OPS-N-16	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 processing of a whole product in granule mode.</p> <p>Used data set : J0</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p> <p>If necessary uncompress recursively input data ( directory (\$WORK_DIR/../../tmp)</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_N_16</code></p> <p>then execute the test shell :</p> <p><code>./lancetest.sh</code></p> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <p><code>cd \$WORK_DIR/OPS/tools/OPStools</code></p> <p><i>Clean.sh</i></p>	Verify that : input, to_pgf and tmp directories are purged.	
<p>3. <i>Launch the OPS in standalone mode:</i></p> <p><code>../../cmd/StartStandAlone.sh 1</code></p> <p><i>\$WORK_DIR N (this step is included in the shell lancetest.sh):</i></p>	The OPS is ready to process input data in nominal mode.	
<p>4. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i></p>	Verify with the	
	<p><b>Is \$WORK_DIR/input/aux_data</b></p> <p>commands that all the input data are available :</p>	

AVHRR product : J0 ( see § 3.5.2)

Other and Stable Parameters , Spectral Database : J0 ( see § 3.5.2)

IASI Context file : J0 (see § 3.5.2) this file is copied from the delivery

Auxiliary data : J0 ( see § 3.5.2)

5. *Copy the processing data, needed to process all granule, in the input directory (using symbolic links)*

Verify with the

**ls \$WORK\_DIR/input**

**ls \$WORK\_DIR/input/product\_model**

**ls \$WORK\_DIR/input/unproc\_data**

commands that all the input data are available :

working order : IASI\_1\_wo\_001 to IASI\_1\_wo\_0041

IASI L0 data : J0 ( see § 3.5.2) ( all granule)

Product model : J0 ( see § 3.5.2) (all granule)

6. Send then a STEP command to the *OPS subsystem* :

*Enter STEP command in the PGF simulator window for all the granules :*

**STEP 1 input/IASI\_1\_wo\_001**

.....

*Wait for the processing end.*

Verify that each STEP command is acknowledged.  
(file CMD\_XXXX)

Verify that with the last granule the processing chain stops

(file CMD\_XXXX)

Verify that the predicted files are produced :

- IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated

- Report file :

**cd output**

**ls**

Verify that the report informations and format are valid.

**cat \*.rpt**

Verify that the HKTM status is send at least every 15 s if there is a progress in the computation

Verify that :

- The last granule is processed :

**cat \*41\*.rpt**

- The verification product size

**ls -l \*VER\***

- The PCD is send using a Log Event

**more LOGxxx**

[\[\[E\\_IFS10\]](#)

[\[E\\_INT9\]](#)

[\[PPF-0500\]](#)

[\[PPF-0060\]](#)

[\[PPF-0080\]](#)

[\[PPF-0510\]](#)

[\[PPF-0530\]](#)

[\[PPF-0691\]](#)

[\[PPF-0530\]](#)

[\[PPF-0820\]](#)

[\[PPF-0550\]](#)

<p>7. Verify the format- of the product files (manual)</p> <p><b>cd output</b></p> <p><b>TraceProduit “ product file name ”</b></p> <p><b>GRANx</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with. A validation tool to verify that the output product are in accordance with the GPFS format. (for exemple , the header and one record of each type)..</p> <p><b>more structure.prod</b> ( is shows 1 MPRH 3 IPR and several GIADR/VIADR/MDR)</p>	<p><a href="#">[E_IFS9]</a></p> <p><a href="#">[PPF-0010]</a></p> <p><a href="#">[PPF-0030]</a></p> <p><a href="#">[E_IFS2ter]</a></p> <p><a href="#">[E_IFS3]</a></p> <p><a href="#">[PPS-3.2-110 pt 5 et 6]</a></p>
<p>8. Verify the structure and the content - of the verification product (manual)</p> <p><b>cd \$WORK_DIR/input/unproc_data</b></p> <p><b>TraceProduit “ L0 product file name ”</b></p> <p><b>GRAN0 (for the first granule)</b></p> <p><b>cd</b></p> <p><b>\$WORK_DIR/OPS/data/TV/GRAN_OPS_N_16/output</b></p> <p><b>TraceProduit “ verification file name ”</b></p> <p><b>GRANx</b></p>	<p>The verification product is dumped together with the L0 file ( binary extraction of VPA, VPB, VPC, VPD, VPE, AP ISP).</p> <p>the binary frames are compared on one line.( dmp file)</p> <p><b>cmp -l \$WORK_DIR</b>  <b>/input/unproc_data/GRAN0/MDR1lum/APVP1.dmp</b>  <b>\$WORK_DIR/OPS/data/TV/GRAN_OPS_N_16/output/ GRANx/MDR_VER_1.dmp</b></p> <p>No differences must appear</p>	<p><a href="#">[[E_IFS2ter]</a></p> <p><a href="#">[E_IFS3]</a></p>

## 5.2.30.GRAN-OPS-N-17\_CORREG

Test Sheet	GRAN-OPS-N-17_CORREG	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the images co registration.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully: execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators: see SIM-OPS-N-01 test.</p> <p>If necessary uncompress recursively input data (directory (\$WORK_DIR/../../tmp)).</p>		
Description :	Expected Result :	Tested Requirement:
<p>9. Connected as iasi_1. You must go in the test specific directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_N_17_CORREG</pre> <p>then execute the test shell :</p> <pre>./lancetest.sh</pre> <p><b>This shell execute automaticaly the following steps (in italic style)</b></p>		
<p>10. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>	Verify that : input, to_pgf and tmp directories are purged.	
<p>11. <i>Launch the OPS in standalone mode:</i></p> <pre>../../cmd/StartStandAlone.sh 1 \$WORK_DIR N (this step is included in the shell lancetest.sh):</pre>	The OPS is ready to process input data in nominal mode.	
<p>12. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i></p>	Verify with the <b>ls \$WORK_DIR/input/aux_data</b> commands that all the input data are available :	



AVHRR product : see work order

Other and Stable Parameters , Spectral Database : see work order

IASI Context file : see work order

Auxiliary data : see work order

13. *Copy the processing data, needed to process all granule, in the input directory (using symbolic links)*

Verify with the

**ls \$WORK\_DIR/input**

**ls \$WORK\_DIR/input/product\_model**

**ls \$WORK\_DIR/input/unproc\_data**

commands that all the input data are available :

working order : IASI\_1\_wo\_001 (20060627054500Z-20060627054800Z)

IASI L0 data : ( all granule) IASI\_xxx\_M02\_20060627054452Z\_20060627054800Z\_N\_T\_20060627070727Z

Product model : (all granule) M02

14. Send then a STEP command to the *OPS subsystem* :

*Enter STEP command in the PGF simulator window for all the granules :*

**STEP 1 input/IASI\_1\_wo\_001**

.....

*Wait for the processing end.*

Verify that each STEP command is acknowledged.  
(file CMD\_xxxx)

Verify that with the last granule the processing chain stops

(file CMD\_xxxx)

Verify that the predicted files are produced :

- IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated
- Report file :

**cd output**

**ls**

Verify that the report informations and format are valid.

**cat \*.rpt**

Verify that the HKTM status is send at least every 15 s if there is a progress in the computation

Verify that :

- The verification product size
- The PCD is send using a Log Event

**more LOGxxx**

<p>15. Verify the format- of the product files (manual)</p> <p><b>cd output</b></p> <p><b>TraceProduit “ product file name ”</b></p> <p><b>GRANx</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with. A validation tool to verify that the output products are in accordance with the GPFS format. (for example , the header and one record of each type)..</p> <p><b>more structure.prod</b> ( is shows 1 MPHR 3 IPR and several GIADR/VIADR/MDR)</p>	
<p>16. Verify the structure and the content - of the verification product (manual)</p> <p><b>cd \$WORK_DIR/input/unproc_data</b></p> <p><b>TraceProduit “ L0 product file name ”</b></p> <p><b>GRAN0 (for the first granule)</b></p> <p><b>cd</b></p> <p><b>\$WORK_DIR/OPS/data/TV/GRAN_OPS_N_17_CORREG/output</b></p> <p><b>TraceProduit “ verification file name ”</b></p> <p><b>GRANx</b></p>	<p>The verification product is dumped together with the L0 file (binary extraction of VPA, VPB, VPC, VPD, VPE, AP ISP).</p> <p>the binary frames are compared on one line.( dmp file)</p> <p><b>cmp -l \$WORK_DIR</b>  <b>/input/unproc_data/GRAN0/MDR1lum/APVP1.</b>  <b>dmp</b>  <b>\$WORK_DIR/OPS/data/TV/GRAN_OPS_N_17_CO</b>  <b>RREG/output/ GRANx/MDR_VER_1.dmp</b></p> <p>No differences must appear for all files produced. The reference data were performed by R.Moreno.</p>	

## 5.2.31.GRAN-OPS-N-18

Test Sheet	GRAN-OPS-N-18	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the the IASI L1 processing:geoloc problems have to raised degraded cases and QIS flags.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p> <p>If necessary uncompress recursively input data ( directory (\$WORK_DIR/../../tmp)).</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_N_18</code></p> <p>then execute the test shell :</p> <p><code>./lancetest.sh</code></p> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <p><code>cd \$WORK_DIR/OPS/tools/OPStools</code></p> <p><code>Clean.sh</code></p>	Verify that : input, to_pgf and tmp directories are purged.	
<p>3. <i>Launch the OPS in standalone mode:</i></p> <p><code>../../cmd/StartStandAlone.sh 1</code></p> <p><code>\$WORK_DIR N (this step is included in the shell lancetest.sh):</code></p>	The OPS is ready to process input data in nominal mode.	
<p>4. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i></p>	Verify with the	
	<p><b>ls \$WORK_DIR/input/aux_data</b></p> <p>commands that all the input data are available :</p>	

AVHRR product : 2 AVHRR files, see work order

Other and Stable Parameters , Spectral Database : see work order

IASI Context file : see work order

IASI\_CTX\_xx\_M02\_20060817100009Z\_XXXXXXXXXXXXXXXXX\_20060117120002Z\_CGS1\_XXXXX00003

Auxiliary data : see work order

<p>5. Copy the processing data, needed to process all granule, in the input directory (using symbolic links)</p>	<p>Verify with the</p> <p><b>ls \$WORK_DIR/input</b></p> <p><b>ls \$WORK_DIR/input/product_model</b></p> <p><b>ls \$WORK_DIR/input/unproc_data</b></p> <p>commands that all the input data are available :</p>	
<p><u>working order</u> : IASI_1_wo_002 (20060817100600-20060817100900)</p> <p><u>IASI L0 data</u> : IASI_XXX_00_M02_20060817100552Z_20060817114437_N_O_20060817114437Z</p> <p><u>Product model</u> : (all granule) M02</p>		
<p>6. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP 1 input/IASI_1_wo_002</b></p> <p>.....</p> <p>Wait for the processing end.</p>	<p>Verify that each STEP command is acknowledged. (file CMD_XXXX)</p> <p>Verify that with the last granule the processing chain stops (file CMD_XXXX)</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> <p><b>cd output</b></p> <p><b>ls</b></p> <p>Verify that the report informations and format are valid.</p> <p><b>cat *.rpt</b></p> <p>Verify that the HKTM status is send at least every 15 s if there is a progress in the computation</p>	
<p>7. Verify the format and flags of the product files (manual)</p> <p><b>cd output</b></p> <p><b>TraceProduit “ product file name ”</b></p> <p><b>GRANx</b></p> <p><b>cd GRANx</b></p>	<p>Each file is dumped with. A validation tool to verify that the output products are in accordance with the GPFS format. Verify the flags and the degraded cases: CG 15 and QIS flag have to be raised (ENG product) .</p>	

## 5.2.32.GRAN-OPS-N-19

Test Sheet	GRAN-OPS-N-19	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 processing when SVM and OSV configuration files are degraded and the management of the “manoeuvre” flags.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p> <p>If necessary uncompress recursively input data ( directory (\$WORK_DIR/../../tmp)).</p>		
Description :	Expected Result :	Tested Requirement:
<p>8. Connected as iasi_1 . You must go in the test specific directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_N_19</pre> <p>then execute the test shell :</p> <pre>./lancetest.sh</pre> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>9. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>	Verify that : input, to_pgf and tmp directories are purged.	
<p>10. <i>Launch the OPS in standalone mode:</i></p> <pre>./../../cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetest.sh):</i></p>	The OPS is ready to process input data in nominal mode.	
<p>11. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i></p>	Verify with the <b>ls \$WORK_DIR/input/aux_data</b> commands that all the input data are available :	

AVHRR product : None

Other and Stable Parameters , Spectral Database : see work order

IASI Context file : see work order

Auxiliary data : see work order

12. *Copy the processing data, needed to process all granule, in the input directory (using symbolic links)*

Verify with the

**ls \$WORK\_DIR/input**

**ls \$WORK\_DIR/input/product\_model**

**ls \$WORK\_DIR/input/unproc\_data**

commands that all the input data are available :

working order : IASI\_1\_wo\_001 (20060910083600Z-20060910083900Z)

IASI L0 data : ( all granule) IASI\_xxx\_00\_M02\_20060910083552Z\_20060910083900Z\_N\_O\_20060910101224Z

Product model : (all granule) M02

13. Send then a STEP command to the *OPS subsystem* :

*Enter STEP command in the PGF simulator window for all the granules :*

**STEP 1 input/IASI\_1\_wo\_001**

.....

*Wait for the processing end.*

Verify that each STEP command is acknowledged.  
(file CMD\_xxxx)

Verify that with the last granule the processing chain stops

(file CMD\_xxxx)

Verify that the predicted files are produced :

- IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated
- Report file :

**cd output**

**ls**

Verify that the report informations and format are valid.

**cat \*.rpt**

Verify that the HKTM status is send at least every 15 s if there is a progress in the computation

14. Verify the format and flags of the product files (manual) <b>cd output</b> <b>TraceProduit “ product file name ”</b> <b>GRANx</b> <b>cd GRANx</b>	Each file is dumped with. A validation tool to verify that the output products are in accordance with the GPFS format. Verify the manoeuvre flags: (level 1C product ) MDR#01 => flag is not raised = 0 MDR#02 => flag is not raised = 0 MDR#03 => flag is raised = 1 MDR#04 => flag is raised = 1 MDR#02 => flag is not raised = 0	

### 5.2.33.GRAI-OPS-N-01

Test Sheet	GRAI-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 intermediate reprocessing mode (1A-1C) of 2 granules.</p> <p>Used data set : J8</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p> <p>If necessary uncompress recursively input data ( directory (\$WORK_DIR/../../tmp)</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 :</p> <p>initialize test environment</p> <p><b>cd &lt;WORKING_ROOT_DIR&gt;</b></p> <p><b>./OPS/tools/OPStools/env_test.sh</b></p> <p>You must go in the test specific directory :</p> <p><b>cd \$WORK_DIR/OPS/data/TV/GRAI-OPS-N-01</b></p> <p>execute the test shell :</p> <p><b>./lancetestbis.sh</b></p> <p>This shell executes automaticaly the followig steps, in italic style</p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <p><b>cd \$WORK_DIR/OPS/tools/OPStools</b></p> <p><b><i>Clean.sh</i></b></p>	<p>Verify that : input, to_pgf and tmp directories are purged:</p> <p><b>cd \$WORK_DIR</b></p> <p><b>ls input</b></p> <p><b>ls to_pgf</b></p> <p><b>ls tmp</b></p> <p><b>ls tmp/data</b></p>	



3. <i>Copy at first, the input auxiliary data, needed to process the whole product in the input directory (using symbolic links)</i>	Verify with the following command that all the input auxiliary data are available :  <b>ls \$WORK_DIR/input/aux_data</b>	
<u>AVHRR product</u> : J0 ( see § 3.5.2) <u>Other and Stable Parameters , Spectral Database</u> : J0 ( see § 3.5.2) <u>IASI Context file</u> : J8 : 2 context files computed with the 2 first granules of J0 (GRAN_OPS_N_16) <u>Auxiliary data</u> : J1 ( see § 3.5.2) without the OBT file		
4. <i>Copy the processing data, needed to process the two granules, in the input directory (using symbolic links)</i>	Verify with the following commands that all the input data are available :  <b>ls \$WORK_DIR/input</b>  <b>ls \$WORK_DIR/input/product_model</b>  <b>ls \$WORK_DIR/input/unproc_data</b>	
<u>working order</u> : IASI_1_wo_001bis and IASI_1_wo_002bis : the processing type is L1b <u>IASI L1A data</u> : J8 : 2 L1A end ENG products computed with the 2 first granules of J0 (GRAN_OPS_N_16) <u>Product model</u> : J1 : ENG, L1B and L1C product models		

<p>5. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p>.....</p> <p>Wait for the processing end.</p>	<p>Verify that each STEP command is acknowledged</p> <p>– during the processing from the directory \$WORK_DIR/tmp with the command :</p> <p><b>tail -f CMD*</b></p> <p>- at the end of the processing from the directory \$WORK_DIR/OPS/data/TV/GRAI_OPS_N_01/output with the command :</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 1 0</p> <p>Verify that the processing chain stops correctly</p> <p>look for ACK START 1 1</p> <p>Verify that the predicted files are produced in the output directory :</p> <p><b>cd \$WORK_DIR/OPS/data/TV</b></p> <p><b>cd GRAI_OPS_N_01/output</b></p> <p><b>ls</b></p> <ul style="list-style-type: none"> <li>• 2 IASI L1B, L1C, ENG product files, named as product model except the processing date which is updated</li> <li>• 2 report files :</li> </ul> <p><b>ls *.rpt</b></p> <p>Verify that the report informations and format are valid.</p> <p><b>nedit *1.rpt</b></p> <p><b>nedit *2.rpt</b></p> <p><b>nedit LOG*</b></p>	<p>[E_IFS8]</p> <p>[E_MOD1]</p> <p>[E_RET1]</p> <p>[PPS-3.2-350]</p> <p>[PPS-3.2-420pt2 et 3]</p> <p>[PPF-0630pt2]</p> <p>[PPF-0060]</p>
--	--	--

<p>6. Verify the content of the product files</p> <p>L1C products must be the same as the product references computed with the 2 first granules of J0 set.</p>	<p>Extract the MDR 1 and 22 of each granule of L1C and ENG products</p> <pre>cd \$WORK_DIR/OPS/data/TV/ cd GRAI_OPS_N_01/output TraceProduit &lt;L1C_PRODUCT&gt; GRAI_&lt;L1C&gt; TraceProduit &lt;ENG_PRODUCT&gt; GRAI_&lt;ENG&gt;</pre> <p>Verify the structure of the L1C products by editing the .prod file produced by the TraceProduit command :</p> <pre>cd \$WORK_DIR/OPS/data/TV/ cd GRAI_OPS_N_01 cd output/GRAI_&lt;L1C&gt; nedit structure.prod cd output/GRAI_&lt;ENG&gt; nedit structure.prod</pre> <p>Use the compfic tool to compare the MDR :</p> <pre>cd \$WORK_DIR/OPS/data/TV/ cd GRAI_OPS_N_01</pre> <p>comparison command:</p> <pre>compfic output/GRAI_&lt;L1C&gt;/MDRx.tra references/output/GRAI_&lt;L1C&gt;/MDRx.tra diff_&lt;L1C&gt;.tra 0.000001 input/precision.tra</pre> <pre>compfic output/GRAI_&lt;ENG&gt;/MDRx.tra references/output/GRAI_&lt;ENG&gt;/MDRx.tra diff_&lt;ENG&gt;.tra 0.000001 input/precision.tra</pre> <p>Verify that the MDR are the same by editing the difference file for each MDR:</p> <pre>cd \$WORK_DIR/OPS/data/TV/ cd GRAI_OPS_N_01 nedit diff_&lt;L1C&gt;.tra nedit diff_&lt;ENG&gt;.tra</pre>	
--	--	--

## 5.2.34.GRAI-OPS-N-02

Test Sheet	GRAI-OPS-N-02	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 intermediate reprocessing mode (1B-1C) of 2 granules.</p> <p>Used data set : J8bis</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p> <p>If necessary uncompress recursively input data ( directory (\$WORK_DIR/../../tmp)</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 :</p> <p>initialize test environment</p> <p><b>cd &lt;WORKING_ROOT_DIR&gt;</b></p> <p><b>./OPS/tools/OPStools/env_test.sh</b></p> <p>You must go in the test specific directory :</p> <p><b>cd \$WORK_DIR/OPS/data/TV/GRAI OPS_N_02</b></p> <p>execute the test shell :</p> <p><b>./lancetestbis.sh</b></p> <p>This shell executes automaticaly the followig steps, in italic style</p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <p><b>cd \$WORK_DIR/OPS/tools/OPStools</b></p> <p><b>Clean.sh</b></p>	<p>Verify that : input, to_pgf and tmp directories are purged:</p> <p><b>cd \$WORK_DIR</b></p> <p><b>ls input</b></p> <p><b>ls to_pgf</b></p> <p><b>ls tmp</b></p> <p><b>ls tmp/data</b></p>	

<p>3. <i>Copy at first, the input auxiliary data, needed to process the whole product in the input directory (using symbolic links)</i></p>	<p>Verify with the following command that all the input auxiliary data are available :</p> <p><b>ls \$WORK_DIR/input/aux_data</b></p>	
<p><u>AVHRR product</u> : J0 ( see § 3.5.2)</p> <p><u>Other and Stable Parameters , Spectral Database</u> : J0 ( see § 3.5.2)</p> <p><u>IASI Context file</u> : J8bis : 2 context files computed with the 2 first granules of J0 (GRAN_OPS_N_16)</p> <p><u>Auxiliary data</u> : J1 ( see § 3.5.2) without the OBT file</p>		
<p>4. <i>Copy the processing data, needed to process the two granules, in the input directory (using symbolic links)</i></p>	<p>Verify with the following commands that all the input data are available :</p> <p><b>ls \$WORK_DIR/input</b></p> <p><b>ls \$WORK_DIR/input/product_model</b></p> <p><b>ls \$WORK_DIR/input/unproc_data</b></p>	
<p><u>working order</u> : IASI_1_wo_001bis and IASI_1_wo_002bis : the processing type is L1c</p> <p><u>IASI L1A data</u> : J8bis : 2 L1B and ENG products computed with the 2 first granules of J0 (GRAN_OPS_N_16)</p> <p><u>Product model</u> : J8bis ( see § 3.5.2) L1C and ENG product models</p>		

<p>5. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granules :</p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p>.....</p> <p>Wait for the processing end.</p>	<p>Verify that each STEP command is acknowledged.</p> <p>- during the processing from the directory \$WORK_DIR/tmp with the command :</p> <p><b>tail -f CMD*</b></p> <p>- at the end of the processing from the directory \$WORK_DIR/OPS/data/TV/GRAI_OPS_N_02/output with the command :</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 1 0</p> <p>Verify that the processing chain stops correctly</p> <p><b>nedit CMD*</b></p> <p>look for ACK START 1 1</p> <p>Verify that the predicted files are produced in the output directory :</p> <p><b>cd \$WORK_DIR/OPS/data/TV</b></p> <p><b>cd GRAI_OPS_N_02/output</b></p> <p><b>ls</b></p> <ul style="list-style-type: none"> <li>• 2 IASI L1C and ENG product files, named as product model except the processing date which is updated</li> <li>• 2 report files :</li> </ul> <p><b>ls *.rpt</b></p> <p>Verify that the report informations and format are valid.</p> <p><b>nedit *1.rpt</b></p> <p><b>nedit *2.rpt</b></p> <p><b>nedit LOG*</b></p>	<p>[E_IFS8]</p> <p>[E_MOD1]</p> <p>[PPS-3.2-350]</p> <p>[PPF-0060]</p>
--	--	--

6. Verify the content- of the product files

L1C products must be the same as the product references computed with the 2 first granules of J0 set.

Extract the MDR 1 and 22 of each granule

**cd \$WORK\_DIR/OPS/data/TV/**

**cd GRAI\_OPS\_N\_02/output**

**TraceProduit <L1C\_PRODUCT>  
GRAI\_<L1C>**

**TraceProduit <ENG\_PRODUCT>  
GRAI\_<ENG>**

Verify the structure of the L1C products by editing the .prod file produced by the TraceProduit command :

**cd \$WORK\_DIR/OPS/data/TV/**

**cd GRAI\_OPS\_N\_02**

**cd output/GRAI\_<L1C>**

**nedit structure.prod**

**cd output/GRAI\_<ENG>**

**nedit structure.prod**

Use the compfic tool to compare the MDR with the reference :

**cd \$WORK\_DIR/OPS/data/TV/**

**cd GRAI\_OPS\_N\_02**

comparison command:

**compfic output/GRAI\_<L1C>/MDRx.tra  
references/outputpur/GRAI\_<L1C>/MDRx.tra  
diff\_<L1C>.tra 0.000001 input/precision.tra**

**compfic output/GRAI\_<ENG>/MDRx.tra  
references/outputpur/GRAI\_<ENG>/MDRx.tra  
diff\_<ENG>.tra 0.000001  
input/precision.tra**

Verify that the MDR are the same by editing the difference file for each MDR:

**cd \$WORK\_DIR/OPS/data/TV/**

**cd GRAI\_OPS\_N\_02**

**nedit diff\_<L1C>.tra**

**nedit diff\_<ENG>.tra**

**OCNES**

**IASI**

**IA-PE-2100-9559-THA**

Edit. : **03**

Date : **21/02/2003**

Rév. : **08**

Date : **28/07/2009**

Référence : -

Page : 154



## 5.2.35.DUMP-OPS-N-01

Test Sheet	DUMP-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate. the IASI L1 processing of a whole product in dump mode</p> <p>Used data set : J1bis</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test specific directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/DUMP_OPS_N_01</pre> <p>then execute the test shell :</p> <pre>./lancetestbis.sh</pre> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>	Verify that : input, to_pgf and tmp directories are purged.	
<p>3. <i>Launch the OPS : in standalone mode:</i></p> <pre>./.../cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetestbis.sh):</i></p>	The OPS is ready to process input data in nominal mode.	
<p>4. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i></p>	Verify with the <b>ls \$WORK_DIR/input/aux_data</b> commands that all the input data are available :	

<u>AVHRR product</u> : J1bis ( see § 4) <u>Other and Stable Parameters , Spectral Database</u> : J1bis ( see § 4) <u>IASI Context file</u> : J1bis (see § 4) this file is copied from the delivery <u>Auxiliary data</u> : J1bis ( see § 4)		
5. <i>Copy the processing data, needed to process the dump, in the input directory (using symbolic links)</i>	Verify with the <b>ls \$WORK_DIR/input</b> <b>ls \$WORK_DIR/input/product_model</b> <b>ls \$WORK_DIR/input/unproc_data</b> commands that all the input data are available :	
<u>working order</u> : IASI_1_wo_001 <u>IASI L0 data</u> : J1bis (see § 4) <u>Product model</u> : J1bis (see § 4)		
6. <i>Send then a STEP command to the OPS subsystem :</i>  <i>Enter STEP command in the PGF simulator window for all the granules :</i>  <b>STEP 1 input/IASI_1_wo_001</b> ..... <i>Wait for the processing end.</i>	Verify that each STEP command is acknowledged. (file CMD_ xxxx)  Verify that the processing chain stops at the end of the dump  Verify that the predicted files are produced : <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> Verify that the report informations and format are valid.  The size of each virtual granule extracted during the processing must be constant and equal to a configuration variable.  <b>cd output</b> <b>grep -I "END OF TASK : ChaineImageISRFEM" TR*</b>  Analyse the number of line <b>X</b> for each granule (in word ChaineImageISRFEM_ <b>X</b> ) which must be constant except for the first and the last granule.	[E_MOD1] [E_MOD3] [PPF-0820] <a href="#">[PPS-3.6-050]</a>
7. Verify the content- of the product files (manual)  <b>cd output</b>	Each file is dumped with. A validation tool to verify that the output product are in accordance with the GPFS format. (for exemple , the header and one record of each type).  TraceProduit " product file name " GRANx	<a href="#">[E_IFS9]</a>

## 5.2.36.OVER-OPS-N-01

Test Sheet	OVER-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 processing concerning the management of the overlap. In this test the beginning date in the L0 is 8 second lower than the date required , (nominal case) . The end of last granulet correspond exactly to the end of a line. The beginning of the first granule doesn't correspond to the beginning of a line .</p> <p>Used data set : J1</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
1. Replay the GRAN-OPS-N-04 test		
2. Verify the content- of the product files	<p>Each file is dumped with. A validation tool to verify that the output product are in accordance with the GPFS format. The goal is to verify the beginning and the end date of each granule.in order to test the overlap mecanism .</p> <ul style="list-style-type: none"> <li>• The date of the last line of each granule must complie with the date of the first line of the following granule ( see the products file name)</li> <li>• The date of the first line of the first granule is not the date of the first ISP in L0 product.( see the product file name)</li> <li>• The date of the last line of the last granule is the date of the last ISP in L0 product. .( see the product file name)</li> </ul>	<p>[E_INT8] [PPF-0640]</p>

## 5.2.37.OVER-OPS-D-02

Test Sheet	OVER-OPS-D-02	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 processing concerning the management of the overlap. In this test the overlap between each granule is lower then 8 s , (degraded case). The first line of the second granule is not included in the product generated.</p> <p>Used data set : J4quar</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 :</p> <p>Initialize test environment</p> <pre>cd &lt;WORKING_ROOT_DIR&gt; ./OPS/tools/OPStools/env_test.sh</pre> <p>You must go in the test specific directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/OVER_OPS_D_02</pre> <p>execute the test shell :</p> <pre>./lancetestbis.sh</pre> <p>This shell executes automatically the followig steps, in italic style</p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>	<p>Verify that : input, to_pgf and tmp directories are purged:</p> <pre>cd \$WORK_DIR</pre> <pre>ls input</pre> <pre>ls to_pgf</pre> <pre>ls tmp</pre> <pre>ls tmp/data</pre>	

<p>3. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i></p>	<p>Verify with the following command that all the input auxiliary data are available :</p> <p><b>ls \$WORK_DIR/input/aux_data</b></p>	
<p><u>AVHRR product</u> : J1 ( see § 3.5.2)</p> <p><u>Other and Stable Parameters , Spectral Database</u> : J1 ( see § 3.5.2)</p> <p><u>IASI Context file</u> : J1 ( see § 3.5.2)</p> <p><u>Auxiliary data</u> : J1 ( see § 3.5.2)</p>		
<p>4. <i>Copy the processing data, needed to process all granules, in the input directory (using symbolic links)</i></p>	<p>Verify with the following commands that all the input data are available :</p> <p><b>ls \$WORK_DIR/input</b></p> <p><b>ls \$WORK_DIR/input/product_model</b></p> <p><b>ls \$WORK_DIR/input/unproc_data</b></p>	
<p><u>working order</u> : IASI_1_wo_001</p> <p><u>IASI L0 data</u> : J4quar ( see § 3.5.2) 4 granules</p> <p><u>Product model</u> : J4quar ( see § 3.5.2) 5 files</p>		
<p>5. <i>Send then a STEP command to the OPS subsystem :</i></p> <p><i>Enter STEP command in the PGF simulator window for each granule:</i></p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p><i>Wait for the processing end.</i></p>	<p>Verify that:</p> <ul style="list-style-type: none"> <li>• each STEP command is acknowledged.</li> </ul> <p>- during the processing from the directory \$WORK_DIR/tmp with the command :</p> <p><b>tail -f CMD*</b></p> <p>- at the end of the processing from the directory \$WORK_DIR/OPS/data/TV/OVER_OPS_D_02/output with the command :</p> <p><b>nedit CMD*</b></p> <ul style="list-style-type: none"> <li>• The last granule is processed.</li> </ul> <p>("ACK STOP" in the file CMD)</p> <ul style="list-style-type: none"> <li>• A logEvent is generated for the second granule due to an absence of line beginning in [t2,t0]</li> </ul> <p><b>cd</b>  <b>\$WORK_DIR/OPS/data/TV/OVER_OPS_D_02/output</b></p> <p><b>more LOG*</b></p> <p>and look for the line that contains the words STARTING and DATE</p>	<p>[PPF-0400pt 1]</p>

<p>6. <i>Verify that the start time of the first line of granule #2 doesn't comply with the end time of the last line of granule #1</i></p>	<p>The start time of the first line of a granule and the end time of the last line are included in the name of the granule :</p> <p>IASI_XXX_1C_M01_&lt;START_TIME&gt;Z_&lt;END_TIME&gt;Z_&lt;NAME_END&gt;</p> <ul style="list-style-type: none"> <li>To verify that a line is missing between the first and the second granule, enter the following commands:</li> </ul> <pre>cd \$WORK_DIR/OPS/data/TV/OVER_OPS_D_02/output ls</pre> <ul style="list-style-type: none"> <li>The start and end time of the L1C granule #1 are 20020808182005 and 20020808182253 while the start and end time of the granule #2 are 20020808182301 and 20020808182557</li> </ul>	<p>[E_INT8bis] [PPF-0610pt1]</p>
---	--	--------------------------------------

**5.2.38.COLD-OPS-N-01**

Test Sheet	COLD-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 processing of a whole product in granule mode in case using a cold initialisation.</p> <p>Used data set : J1 and GRAN_N_04 directory</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:

<p>1. Connected as iasi_1 .</p> <p>initialize test environment</p> <pre>cd &lt;WORKING_ROOT_DIR&gt; ./OPS/tools/OPStools/env_test.sh</pre> <p>You must go in the test specific directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/GRAN_OPS_N_04</pre> <p>modify the shell script to process only the first work order:</p> <p><b>nedit lancetestbis.sh</b></p> <p>look for the line</p> <pre>liste_wo=`ls input/*wo*  egrep -v bis`</pre> <p>and replace it with</p> <pre>liste_wo=`ls input/*wo_001  egrep -v bis`</pre> <p>look for the line</p> <pre>ls input/*wo*  egrep -v bis&gt;tmp</pre> <p>and replace it with</p> <pre>ls input/*wo_001  egrep -v bis&gt;tmp</pre> <p>save the modifications</p> <p>then execute the test shell :</p> <pre>./lancetestbis.sh</pre> <p>This shell executes automaticaly the followig steps, in italic style</p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>	<p>Verify that : input, to_pgf and tmp directories are purged:</p> <pre>cd \$WORK_DIR ls input ls to_pgf ls tmp ls tmp/data</pre>	
<p>3. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i></p>	<p>Verify with the</p> <pre>ls \$WORK_DIR/input/aux_data</pre> <p>commands that all the input data are available :</p>	



AVHRR product : J1 ( see § 3.5.2) same as GRAN\_N\_04

Other and Stable Parameters , Spectral Database : J1 ( see § 3.5.2) same as GRAN\_N\_04

IASI Context file ( cold context): \$WORK\_DIR/OPS/data/input/aux\_data/AUX/

IASI\_CTX\_xx\_M01\_20021008025258Z\_XXXXXXXXXXXXXXXXX\_20021008165703Z\_IASI\_XXXXX01001\_1

Auxiliary data : J1 ( see § 3.5.2) same as GRAN\_N\_04

Working order : IASI\_1\_wo\_001 same as GRAN\_N\_04 that must include the cold context file

IASI L0 data : J1 ( see § 3.5.2) ( first granule) same as GRAN\_N\_04

Product model : J1 ( see § 3.5.2) ( first granule) same as GRAN\_N\_04

<p>4. <i>Send then a STEP command to the OPS subsystem :</i></p> <p><i>Enter STEP command in the PGF simulator window for each granule:</i></p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p><i>Wait for the processing end.</i></p>	<p>Verify that the STEP command is acknowledged.</p> <p><b>more</b></p> <p><b>\$WORK_DIR/OPS/data/TV/GRAN_OPS_N_04/output/CMD*</b></p> <p>verify that ACK STEP &lt;N&gt; 0 is written where N is the work order number</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> <p><b>cd output</b></p> <p><b>ls</b></p> <p>Verify that the report informations and format are valid.</p> <p><b>cat *001.rpt</b></p>	
<p>5. <i>Verify the content- of the product files</i></p>	<p>Dump the context file with the following commands :</p> <p><b>cd \$WORK_DIR/OPS/data/TV/</b></p> <p><b>cd GRAN_OPS_N_04/output</b></p> <p><b>TraceContexte &lt;CONTEXTE_FILE&gt;</b></p> <p><b>Contexte.tra</b></p> <p>in the context file, the BBT written must show that for the first line of the granule, the BBT used is the same as for the second line: BBT(ln-1) :</p> <p>look for "BIMSBBTHist" in Contexte.tra</p>	

## 5.2.39.LOG-OPS-N-01

Test Sheet	LOG-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the use of log event message during. the IASI L1 processing of a granule in case of various situations.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1, you must go in the test specific directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/LOG_OPS_N_01</pre> <p>then execute the test shell :</p> <pre>.. /lancetestbis.sh</pre> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>		
<p>3. <i>Copy at first, the input auxiliary data, needed to process both granules in the input directory (using symbolic links)</i></p>	<p>Verify with the</p> <pre>ls \$WORK_DIR/input/aux_data</pre> <p>command that all the input data are available :</p>	
<p><u>AVHRR product</u> : J1 (see § 3.5.2)</p> <p><u>Other and Stable Parameters , Spectral Database</u> : J1 (see § 3.5.2)</p> <p><u>IASI Context file</u> : J1 (see § 3.5.2) this file is copied from the delivery</p> <p><u>Auxiliary data</u> : J1 (see § 3.5.2)</p>		

4. Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)	Verify with the <b>ls \$WORK_DIR/input</b> <b>ls \$WORK_DIR/input/product_model</b> <b>ls \$WORK_DIR/input/unproc_data</b> commands that all the input data are available :	
<u>working order</u> : IASI_1_wo_001 <u>IASI L0 data corrupted</u> : TV_THA_IA0_1 ( see § 3.5.2) (J1quart) <u>Product model</u> : J1 (see § 3.5.2) (first granule)		
5. Send then a STEP command to the OPS subsystem :  <b>STEP 10 input/IASI_1_wo_001</b>  Wait for the processing end.	Verify that the STEP command is accepted. Verify that the predicted files are produced and that: <ul style="list-style-type: none"> <li>Several log events must exist, describing an error during the reading of input data</li> </ul> <b>cd output</b> <b>more LOG*</b> Verify that the message is described in the user manual document	[PPF-0400 pt 2,3,4] [PPS-3.2-100] [SYS-0480] [E_INT11] [E_INT13] [PPF-0910]
6. Copy the processing data, needed to process the second granule, in the input directory	Verify with the <b>ls \$WORK_DIR/input</b> <b>ls \$WORK_DIR/input/product_model</b> <b>ls \$WORK_DIR/input/unproc_data</b> commands that all the input data are available :	
<u>working order</u> : IASI_1_wo_002 <u>inexisting IASI L0 data</u> <u>Product model</u> : J1 ( see § 3.5.2) (second granule)		
7. Send then a STEP command to the OPS subsystem :  <b>STEP 2 input/IASI_1_wo_002</b>	Verify that the STEP command is rejected. Verify that the predicted files are not produced and that : <ul style="list-style-type: none"> <li>No report file is produced</li> <li>A log event must describe the problem (input data not found)</li> </ul> <b>cd output</b> <b>more LOG*</b> Verify that the message is described in the user manual document	[PPF-0400 pt 4] [PPS-3.2-100] [SYS-0480] [E_INT11] [E_INT13] PPF-0910]
8. Stop the OPS : send a STOP command to the OPS subsystem :  <b>STOP 20</b>	Verify that the output data are still available after the OPS is stopped.	

**5.2.40.ERR-OPS-N-01**

Test Sheet	ERR-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to test the behaviour of the OPS in case of abnormal end of an applicative process.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version is install successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
1. Launch the OPS : Execute the 3 first steps of RUN-OPS-N-01 test.	The OPS is ready to process input data.	
2. Simulate an abnormal end of the SD process <b>kill -11 &lt;SD process id&gt;</b>	Verify with the <b>ps -ef   grep MainProcess</b> <b>ps -ef   grep Serveur</b> commands that all the OPS process are stopped. Verify that appropriate log events are generated in order to inform MCS subsystem.	[PPF-0430] [PPF-0550]
3. Clean the OPS validation environment : <b>cd \$WORK_DIR/OPS/tools/OPStools</b> <b>Clean.sh</b>	Verify that : input, to_pgf and tmp directories are purged.	

## 5.2.41.CMD-OPS-N-01

Test Sheet	CMD-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the processing of PGF commands. [PPS-3.2-010]</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version is install successfully : execution of INST-OPS-N-02 test.</p> <p>the test is executed in CGS simulated environment, so it is requested to start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
1. Launch the OPS in normal mode: Execute the first 3 steps of RUN-OPS-N-01 test.	The OPS is ready to process input data in nominal mode.	
2. Connected as iasi_1, copy the OPS input processing data, needed to process the first, the middle and the last granules, in the input directory:  <b>cd \$WORK_DIR/OPS/tools/OPStools</b> <b>copyTestData.sh CMD_OPS_N_01</b> <b>input/aux_data</b>	Verify with the  <b>ls \$WORK_DIR/input/aux_data</b> command that all the input aux data are available :	[PPS-3.2-110 pt1,2,3,4,7]
<p><u>AVHRR product</u> : J10 (see § 3.5.2)</p> <p><u>Other and Stable Parameters , Spectral Database</u> : J1 (see § 3.5.2)</p> <p><u>IASI Context file</u> : J1 (see § 3.5.2)</p> <p><u>Auxiliary data</u> : J1 (see § 3.5.2)</p>		

<p>3. Copy the OPS input processing data, needed to process the first, the middle and the last granules, in the input directory:</p> <pre>copyTestData.sh CMD_OPS_N_01 input/product_model  copyTestData.sh CMD_OPS_N_01 input/unproc_data  cp \$WORK_DIR/OPS/data/TV/CMD_OPS_N_01/input/IASI_1_wo_001bis \$WORK_DIR/input/.  cp \$WORK_DIR/OPS/data/TV/CMD_OPS_N_01/input/IASI_1_wo_002bis \$WORK_DIR/input/.  cp \$WORK_DIR/OPS/data/TV/CMD_OPS_N_01/input/IASI_1_wo_003bis \$WORK_DIR/input/.</pre>	<p>Verify with the</p> <pre>ls \$WORK_DIR/input ls \$WORK_DIR/input/product_model ls \$WORK_DIR/input/unproc_data</pre> <p>commands that all the input data are available :</p>	<p>[PPS-3.2-110 pt1,2,3,4,7]</p>
<p><u>working order</u> : IASI_1_wo_001, IASI_1_wo_002 and IASI_1_wo_003</p> <p><u>IASI L0 data</u> : J10 ( see § 3.5.2) (two granules)</p> <p><u>Product model</u> : J10 ( see § 3.5.2) (five files)</p>		
<p>4. Send 3 STEP commands to the OPS subsystem :</p> <pre>STEP 1 input/IASI_1_wo_001bis STEP 2 input/IASI_1_wo_002bis STEP 3 input/IASI_1_wo_003bis</pre>	<p>For each STEP command, a CODED acknowledgement is sent to the PGF synchronously with the associated log event to the MCS</p> <p>The OPS starts to process the first granule.</p>	<p>[SYS-0440] [SYS-0490] [IF-ALGO-4.2-140] [PPF-0350]</p>
<p>5. Send a SUSPEND command for the current processing:</p> <pre>SUSPEND 13</pre>	<p>A CODED acknowledgement is sent to the PGF synchronously with the associated log event to the MCS.</p> <p>When the processing is suspended, a COMPLETED acknowledgement is sent to the PGF synchronously with the associated log event to the MCS.</p> <pre>cd \$WORK_DIR/tmp cat CMD*</pre>	<p>[SYS-0390] [SYS-0440] [SYS-0490] [IF-ALGO-4.2-140] [PPS-3.2-030] [IF-ALGO-4.2-100]</p>

<p>6. Send RESUME command: <b>RESUME 14</b></p> <p>Wait for the end of the processing.</p>	<p>A CODED acknowledgement is sent to the PGF synchronously with the associated log event to the MCS.</p> <p>When the processing is restarted, a COMPLETED acknowledgement is sent to the PGF synchronously with the associated log event to the MCS</p> <p>Verify that the OPS restarts the suspended processing.</p> <p><b>cd \$WORK_DIR/tmp</b> <b>cat LOG*</b></p>	<p>[SYS-0390] [SYS-0440] [SYS-0490] [IF-ALGO-4.2-140] [PPS-3.2-030] [IF-ALGO-4.2-100]</p>
<p>7. While processing the second granule, send a BREAK command: <b>BREAK 15</b></p>	<p>A CODED acknowledgement is sent to the PGF synchronously with the associated log event to the MCS in less than 15 s.</p> <p>Verify that the current processing is stopped. A report is provided in order to warn the PGF that the granule processing is stopped.</p> <p>When the break is completed, a COMPLETED acknowledgement is sent to the PGF synchronously with the associated log event to the MCS in less than 1mn.</p> <p>The OPS starts to process last granule.</p> <p><b>cd \$WORK_DIR/tmp</b> <b>cat CMD*</b> <b>cat LOG*</b></p>	<p>[SYS-0390] [SYS-0440] [SYS-0490] [IF-ALGO-4.2-140] <a href="#">[PPF-0681]</a> <a href="#">[PPF-0693]</a></p>
<p>8. Send a BREAK command to stop the last granule processing: <b>BREAK 16</b></p>	<p>A CODED acknowledgement is sent to the PGF synchronously with the associated log event to the MCS</p> <p>Verify that the current processing is stopped. A report is provided in order to warn the PGF that the granule processing is stopped.</p> <p>When the break is completed, the OPS shuts down.</p> <p><b>cd \$WORK_DIR/tmp</b> <b>cat CMD*</b> <b>cat LOG*</b></p>	<p>[SYS-0440] [SYS-0490] [IF-ALGO-4.2-140] <a href="#">[PPF-0681]</a></p>
<p>9. Clean the OPS validation environment : <b>cd \$WORK_DIR/OPS/tools/OPStools</b> <b>Clean.sh</b></p>	<p>Verify that : input, to_pgf and tmp directories are purged:</p> <p>ls \$WORK_ROOT_DIR/input ls \$WORK_ROOT_DIR/to_pgf ls \$WORK_ROOT_DIR/tmp</p>	

7. Stop the PGF simulator , enter : <b>PGFS_ABORT</b>	The PGF simulator is stopped.	
--	-------------------------------	--



## 5.2.42.HKTM-OPS-N-01

Validé par les tests RUN-OPS-N-01 et GRAN-OPS-N-01 de l'OPS : [\[PPF-0390pt1\]](#)

### 5.2.43.PERF-OPS-N-01

Test Sheet	PERF-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the. IASI L1 CPU performance of a whole product in granule mode</p> <p>Used data set : J1</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p> <p>If necessary uncompress recursively input data ( directory (\$WORK_DIR/../../tmp)</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/PERF OPS_N_01</code></p> <p>then execute the test shell :</p> <p><code>./lancetestbis.sh</code></p> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. Verifie that the processing chain is working alone on the configuration using the command</p> <p><code>“ps -ef   more”</code></p>	The OPS is alone ( no other user process working)	
<p>3. <i>Clean the OPS validation environment :</i></p> <p><code>cd \$WORK_DIR/OPS/tools/OPStools</code></p> <p><i>Clean.sh</i></p>	Verify that : input, to_pgf and tmp directories are purged.	
<p>4. <i>Launch the OPS :</i></p> <p><i>(this step is included in the shell lancetestbis.sh)</i></p>	The OPS is ready to process input data in nominal mode.	

<p>5. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i></p>	<p>Verify with the</p> <p><b>ls \$WORK_DIR/input/aux_data</b></p> <p>commands that all the input data are available :</p>	
<p><u>AVHRR product</u> : J1 ( see § 3.5.2)</p> <p><u>Other and Stable Parameters , Spectral Database</u> : J1 ( see § 3.5.2)</p> <p><u>IASI Context file</u> :</p> <p><u>Auxiliary data</u> : J1 ( see § 3.5.2)</p>		
<p>6. <i>Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</i></p>	<p>Verify with the</p> <p><b>ls \$WORK_DIR/input</b></p> <p><b>ls \$WORK_DIR/input/product_model</b></p> <p><b>ls \$WORK_DIR/input/unproc_data</b></p> <p>commands that all the input data are available :</p>	
<p><u>working order</u> : IASI_1_wo_001</p> <p><u>IASI L0 data</u> : J1 and J10 ( see § 3.5.2) (five granules)</p> <p><u>Product model</u> : J1 ( see § 3.5.2) (five granules)</p>		
<p>7. <i>Send then a STEP command to the OPS subsystem :</i></p> <p><i>Enter STEP command in the PGF simulator window for all the granules :</i></p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p>.....</p> <p><i>Wait for the processing end.</i></p>	<p>Verify that each STEP command is acknowledged. (file CMD_xxxx)</p> <p>Verify that with the last granule the processing chain stops</p> <p>Verify that the predicted files are produced :</p> <ul style="list-style-type: none"> <li>• Report file :</li> </ul> <p>Verify that the report informations and format are valid.</p> <p><b>cd output</b></p> <p><b>cat *t</b></p>	
<p>8. Verify the performances for the processing of each granule</p>	<p>For each step, the elapsed time is printed and must be lower than 22*8s, except for the first granule due to the initialisation stage</p> <p>For the last granule which contains 10 lines the performance must be proportionnal</p> <p>NB ; cpu usage comes from the report order file</p> <p><b>cd output</b></p> <p><b>grep -i elapsedtime *t</b></p>	<p><a href="#">[PPF-0200]</a></p> <p><a href="#">[E_CTX2]</a></p> <p><a href="#">[PPF-0210]</a></p>

## 5.2.44.PERF-OPS-N-02

Test Sheet	PERF-OPS-N-02	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 CPU performance of a whole product in granule mode over a long period (several orbits).</p> <p>Used data set : J5</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p> <p>If necessary uncompress recursively input data ( directory (\$WORK_DIR/../../tmp)</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/PERF OPS N_02</code></p> <p>then execute the test shell :</p> <p><code>./lancetestbis.sh</code></p> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. Verifie that the processing chain is working alone on the configuration using the command</p> <p>“ps -ef   more”</p>	The OPS is alone ( no other user process working)	
<p>3. <i>Clean the OPS validation environment :</i></p> <p><code>cd \$WORK_DIR/OPS/tools/OPStools</code></p> <p><i>Clean.sh</i></p>	Verify that : input, to_pgf and tmp directories are purged.	
<p>4. <i>Launch the OPS :</i></p> <p><i>(this step is included in the shell lancetestbis.sh)</i></p>	The OPS is ready to process input data in nominal mode.	

5. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i>	Verify with the <b>ls \$WORK_DIR/input/aux_data</b> commands that all the input data are available :	
<u>AVHRR product</u> : J1 ( see § 3.5.2) <u>Other and Stable Parameters , Spectral Database</u> : J1 ( see § 3.5.2) <u>IASI Context file</u> : J1 (see § 3.5.2) this file is copied from the delivery <u>Auxiliary data</u> : J1 ( see § 3.5.2)		
6. <i>Copy the processing data, needed to process the first granule, in the input directory (using symbolic links)</i>	Verify with the <b>ls \$WORK_DIR/input</b> <b>ls \$WORK_DIR/input/product_model</b> <b>ls \$WORK_DIR/input/unproc_data</b> commands that all the input data are available :	
<u>working order</u> : IASI_1_wo_001 <u>IASI L0 data</u> : J5 ( see § 3.5.2) ( all granule in 2 orbit) <u>Product model</u> : J5 ( see § 3.5.2) ( all granule in 2 orbit)		
7. <i>Send then several STEP command to the OPS subsystem :</i>  <i>Enter STEP command in the PGF simulator window for all the granules of the two dump:</i> <b>STEP x input/IASI_x_wo_001</b> ..... <i>Wait for the processing end.</i>	Verify that each STEP command is acknowledged. Verify that with the last granule the processing chain stops Verify that the predicted files are produced : <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul> Verify that the report informations and format are valid.	[PPF-530-pt2]
8. <i>Replay step 4 to 7 using the second orbit and the context generates at the end of the first orbit</i>	. For each step, the elapsed time is printed in the report file and must be coherent with performance requirements (8s for each line processed)	[PPS-3.5-035pt1] [PPS-3.6-040]
9. <i>Replay step 4 to 7 using the third orbit and the context generates at the end of the second orbit</i>	. For each step, the elapsed time is printed in the report file and must be coherent with performance requirements (8s for each line processed)	[PPS-3.5-035pt1] [PPS-3.6-040]

10. Verify the performances for the processing of each granule	<p>For each step, the elapsed time is printed and must be coherent with performance requirements</p> <p><b>cd output</b></p> <p><b>grep -i elapsedtime *t</b></p> <p>For the whole orbit the processing time must be stable and the software must mention any problems</p>	<a href="#">[PPS-3.5-035pt1]</a> <a href="#">[PPS-3.6-040]</a>
--	--	---

## 5.2.45.ALGO-OPS-N-01

Test Sheet	ALGO-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing in granule mode</p> <p>Used data set : J1</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p>		
Description :	Expected Result :	Tested Requirement:
1. Replay the GRAN-OPS-N-04 test (Execute the first 5 steps)		
2. <i>Verify the content- of the product files.(automatic) (1C and ENG)</i>  <b>cd output</b>  <b>TraceProduit “product file name ”</b> <b>GRANx AUTO</b>	Each file is dumped with. A validation tool. (, the two first granule of L1C product ).  Dump all the lines of the two first granules (NB : the reference files must have been dumped before)  <b>cd references</b> <b>dump.ksh</b>	[IF-ALGO-4.4-131] [IF-ALGO-4.4-140] [IF-ALGO-4.4-160] [IF-ALGO-4.4-170] [IF-ALGO-4.4-180] [IF-ALGO-4.4-183]

<p>3. <i>Compare the content- of the product files (automatic)</i></p> <p><i>compfic ../references/GRANx/MDRx.tra GRANx/MDRx.tra GRANx/MDRx.tra.diff\$prec ../input/precision.tra</i></p>	<p>The previous files can be compared with references files using the “compfic” tool.</p> <p>Analyse the number of differences ( key word NBDIFFBALEUR) in file <b>MDRx.tra.diff. or VIADRxx.tra.diff</b> in each directory output/GRANx</p> <p>The tool must assume no differences for specific data ( MDR or VIADR):</p> <ul style="list-style-type: none"> <li>• Spectra (MDR) (1C product ), Key word <b>SPECTRE</b></li> <li>• image (MDR)(1C product ), Key word <b>IMAGE</b></li> </ul> <p>(V1 bis only for the following data)</p> <ul style="list-style-type: none"> <li>• spectral shifts and associated quality (MDR) (ENG product ) (SSD), Key words <b>ENG_GSSDWNShift and ENG_GSSDWNShiftQUAL</b></li> <li>• position of the interferometric axis ( before and after filtering) (MDR) (ENG product) (IAX, FAX), Key words <b>ENG_GIAXAXE and ENG_GFAXAXE</b></li> </ul> <p>spectral database interpolation (MDR) (1C product ) (ISF), Key words <b>N1C_GISFLINORIGIN and N1C_GISFCOLORIGIN and N1C_GISFPDS</b> <li>• extremum spectra (VIADR) (ENG product) (EXS), Key word <b>VIADRENG_MEXSS</b></li> <li>• bad pixels survey (VIADR and MDR ) (ENG product ) (DPT), Key words <b>VIADRENG_MDPTIISBADHEALTHPIX and ENG_MDPTVARIMAG</b></li> <li>• radiometric calibration noise (MDR) (ENG product ),(MCX), Key words <b>ENG_MMCTXNOISECALRAD and ENG_MMCTXBIASCALRAD</b></li> <li>• spectra under and overflow (MDR) (ENG product ) (DOC), Key words <b>ENG_GDOCPOSOVERFLOW</b></li> </p>	<p><a href="#">[E_ENC4]</a> [PPF-0340]</p>
---	---	--



(following analysis for V2.1)

- co-localisation IASI-AVHRR (MDR) (1C product) (IAC),

key-words `ENG_GIACOFFSETIISAVHRR` and `ENG_GCCSOFFSETSONDIIS`

- localisation (MDR) (1C product) (GEO),

key-word `N1C_GGEOSONDLOC`

- 41\_CCS output (MDR) (1C product) (CCS),

key-words `N1C_GCCSRADANALX(Y)` and `N1C_GCCSRADANALMEAN(STD)`

for each data type a relative precision is choosen and is used to compare the results(file input/precision.tra).

The data gap caused by a calibration mode on a non earth sight is not computed an the ISRFEM axis is filtered without using these data ( see the context file)

## 5.2.46.ALGO-OPS-N-02

Test Sheet	ALGO-OPS-N-02	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing of IIS images in case of various coding type (8, 10, 12 bits)</p> <p>Used data set : J7, J7bis, J7ter</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Launch the OPS :</p> <p>Execute the first 3 steps of RUN-OPS-N-01 test.</p>	The OPS is ready to process input data in nominal mode.	
<p>2. Connected as iasi_1 then go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/ALGO_OPS_N_02</code></p> <p>then execute the test shell :</p> <p><b><code>./lancetestbis.sh</code></b></p> <p>This shell executes automaticaly the following steps, in italic style</p>		

<p>3. <i>Copy(using symbolic links) all data needed to process both granules in the directories</i>  <i>\$WORK_DIR/input/aux_data</i>  <i>\$WORK_DIR/input/unproc_data</i>  <i>\$WORK_DIR/input/product_model</i></p>	<p>Data are copied in  <b>\$WORK_DIR/input/aux_data</b>  <b>\$WORK_DIR/input/unproc_data</b>  <b>\$WORK_DIR/input/product_model</b></p> <p>A screen message confirms the operation :  <i>mise en place des données auxiliaires</i>  <i>Copy SUCCESS</i>  <i>END</i></p> <p><i>mise en place des product model</i>  <i>Copy SUCCESS</i>  <i>END</i></p> <p><i>mise en place des données d'entrée</i>  <i>Copy SUCCESS</i>  <i>END</i></p>	
<p><u>AVHRR product</u> : J1 ( see § 3.5.2)</p> <p><u>Other and Stable Parameters , Spectral Database</u> : J1 ( see § 3.5.2)</p> <p><u>IASI Context file</u> : J1 (see § 3.5.2) this file is copied from the delivery</p> <p><u>Auxiliary data</u> : J1 ( see § 3.5.2)</p>		
<p><u>working order</u> : IASI_1_wo_001bis</p> <p><u>IASI L0 data</u> : J7 ( see § 3.5.2) ( one granule)</p> <p><u>Product model</u> : J7 ( see § 3.5.2) ( one granule)</p>		
<p>4. <i>Send then a STEP command to the OPS subsystem :</i></p> <p><i>Enter STEP command in the PGF simulator window :</i></p> <p><i>STEP 1 input/IASI_1_wo_001bis</i></p> <p><i>Wait for the processing end.</i></p>	<p>Verify that the STEP command is acknowledged.</p> <p>– during the processing from the directory  \$WORK_DIR/tmp with the command :</p> <p><b>tail -f CMD*</b></p> <p>- at the end of the processing from the directory  \$WORK_DIR/OPS/data/TV/ALGO_OPS_N_02/output with the command :</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 1 0</p> <p>Verify the process terminates correctly</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 1 1</p> <p>look for ACK START 1 3</p> <ul style="list-style-type: none"> <li>• Verify after step 7 that  Product files are generated  Report files are generated and notify no error</li> </ul>	

working order : IASI\_1\_wo\_002bis

IASI L0 data : J7bis ( see § 3.5.2) ( one granule)

Product model : J7bis ( see § 3.5.2) ( one granule)

<p>5. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window :</p> <p><b>STEP 1 input/IASI_1_wo_002bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is acknowledged.</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 2 0</p> <p>Verify the process terminates correctly</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 2 1</p> <p>look for ACK START 1 3</p> <ul style="list-style-type: none"> <li>• Verify after step 15 that</li> </ul> <p>Produt files are generated</p> <p>Report files are generated and notify no error</p>	
<p><u>Working order</u> : IASI_1_wo_003bis</p> <p><u>IASI L0 data</u> : J7ter ( see § 3.5.2) ( one granule)</p> <p><u>Product model</u> : J7ter ( see § 3.5.2) ( one granule)</p>		
<p>6. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window :</p> <p><b>STEP 1 input/IASI_1_wo_003bis</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is acknowledged.</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 3 0</p> <p>Verify the process terminates correctly</p> <p><b>nedit CMD*</b></p> <p>look for ACK STEP 3 1</p> <p>look for ACK START 1 3</p> <ul style="list-style-type: none"> <li>• Verify after step 15 that</li> </ul> <p>Produt files are generated</p> <p>Report files are generated and notify no error</p>	
<p>7. Dump the content- of the product files :</p> <p><b>dumpeprod.sh</b></p>	<p>Each product file is dumped with a validation tool (TraceProduit) to verify that the output product are in accordance with the GPFS format.</p> <p>Verify that 3 directories :</p> <p>\$WORK_DIR/output/GRAN_N1C_1</p> <p>\$WORK_DIR/output/GRAN_N1C_2</p> <p>\$WORK_DIR/output/GRAN_N1C_3</p> <p>are present and contains 22 MDRx.tra files</p>	

<p>8. Dump the content- of the reference files :</p> <p><b>dumperef.sh</b></p>	<p>Each reference file is dumped with a validation tool (TraceProduit) in order to be compared with the product généré by the OPS.</p> <p>Verify that 3 directories :</p> <p>\$WORK_DIR/reference/J7/GRAN_N1C_1</p> <p>\$WORK_DIR/reference/J7bis/GRAN_N1C_2</p> <p>\$WORK_DIR/reference/J7ter/GRAN_N1C_3</p> <p>are present and contains 22 MDRx.tra files</p>	
<p>9. Compare the content- of the product files</p> <p><b>compare.sh</b></p>	<p>Dumped product files are compared with dumped references files using the “compfic” tool.</p> <p>For each data type a relative precision is choosen (1e-6) and is used to compare the results.</p> <p>Results are saved in xxx.diff files in</p> <p>\$WORK_DIR/output/GRAN_N1C_1</p> <p>\$WORK_DIR/output/GRAN_N1C_2</p> <p>\$WORK_DIR/output/GRAN_N1C_3</p> <p>Verify relative errors are acceptable</p>	<p><a href="#">[E_INT2]</a></p>

## 5.2.47.ALGO-OPS-N-03

Test Sheet	ALGO-OPS-N-03	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 numerical results in case of a modification of the instrument mode (calibration mode)</p> <p>Used data set : J10</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p>		
Description :	Expected Result :	Tested Requirement:
1. Replay the GRAN-OPS-N-08 test (Execute the first steps)		
2. Verify the content of the product files .(automatic) <i>cd output</i> <i>TraceProduit “ product file name ”</i> <i>GRANx</i>	Dump the three first granules of L1C product with the validation tool TraceProduit :  <b>TraceProduit “ product file name ” GRANx AUTO</b>  Dump all the lines of the g1,g2,g3 granule  (NB : the reference files must have been dumped before)	
3. <i>cd GRANx</i>		

4. *Compare the content of the product files  
(automatic)*

*compfic MDRx.tra*  
*.././references/GRANx/MDRx*  
*MDRx.tra.diff 10 .././input/précision.tra*

The previous files can be compared with references files using the “compfic” tool.

Analyse the number of differences ( key word NBDIFFVALEUR) in file **MDRx.tra.diff. or VIADRxx.tra.diff** in each directory output/GRANx

The tool must assume no differences for specific data ( MDR or VIADR) :

- Spectra (MDR) (1C product )  
Key word **SPECTRE**
- image (MDR)(1C product )  
Key word **IMAGE**

(V1 bis only for the following data)

- spectral shifts and associated quality (MDR) (ENG product ) (SSD)  
Key words **ENG\_GSSDWNSHIFT** and **ENG\_GSSDWNSHIFTQUAL**
- position of the interferometric axis ( before and after filtering) (MDR) (ENG product) (IAX, FAX)  
Key words **ENG\_GIAXAXE** and **ENG\_GFAXAXE**
- spectral database interpolation (MDR) (1C product ) (ISF)  
Key words **N1C\_GISFLINORIGIN** and **N1C\_GISFCOLORIGIN** and **N1C\_GISFPDS**
- extremum spectra (VIADR) (ENG product) (EXS)  
Key word **VIADRENG\_MEXSS**
- bad pixels survey (VIADR and MDR ) (ENG product ) (DPT),  
Key words **VIADRENG\_MDPTIISBADHEALTHPIX** and **ENG\_MDPTVARIMAG**
- radiometric calibration noise (MDR) (ENG product ),(MCX),  
Key words **ENG\_MMCXNOISECALRAD** and **ENG\_MMCXBIASCALRAD**

[\[E\\_INT10\]](#)

[\[E\\_SEQ1\]](#)

[\[E\\_SEQ2\]](#)

[\[E\\_SEQ4\]](#)

[\[E\\_SEQ5\]](#)

[\[E\\_DEG3\]](#)

[\[E\\_MOD2\]](#)

- spectra under and overflow (MDR)  
(ENG product ) (DOC)  
Key words **ENG\_GDOCPOSOVERFLOW**  
(following analysis for V2.1)
- co-localisation IASI-AVHRR (MDR)  
(1C product) (IAC),  
key-words **ENG\_GIACOFFSETIISAVHRR**  
and **ENG\_GCCSOFFSETSONDIIS**
- localisation (MDR) (1C product)  
(GEO)  
key-word **N1C\_GGEOSONDLOC**
- 41\_CCS output (MDR) (1C product)  
(CCS)  
key-words **N1C\_GCCSRADANALX(Y)** and  
**N1C\_GCCSRADANALMEAN(STD)**

for each data type a relative precision is choosen  
and is used to compare the results(file  
input/precision.tra).

The data gap caused by a calibration mode on a  
non earth sight is not computed and the ISRFEM  
axis is filtered without using these data ( see the  
context file)



## 5.2.48.ALGO-OPS-N-04

Test Sheet	ALGO-OPS-N-04	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 numerical results in case of specific data (non nominal cases in algorithms )</p> <p>Used data set : J13</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Launch the OPS :</p> <p>Execute the first 3 steps of RUN-OPS-N-01 test.</p>	The OPS is ready to process input data in nominal mode.	
<p>2. Connected as iasi_1 then go in the test specific directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/ALGO_OPS_N_04</code></p> <p>then execute the test shell :</p> <p><b><code>./lancetestbis.sh</code></b></p> <p>This shell executes automaticaly the following steps, in italic style</p>		

<p>3. <i>Copy(using symbolic links) all data needed to process both granules in the directories</i>  <i>\$WORK_DIR/input/aux_data</i>  <i>\$WORK_DIR/input/unproc_data</i>  <i>\$WORK_DIR/input/product_model</i></p>	<p>Data are copied in  <b>\$WORK_DIR/input/aux_data</b>  <b>\$WORK_DIR/input/unproc_data</b>  <b>\$WORK_DIR/input/product_model</b></p> <p>A screen message confirms the operation :  <i>mise en place des données auxiliaires</i>  <i>Copy SUCCESS</i>  <i>END</i>  <i>mise en place des product model</i>  <i>Copy SUCCESS</i>  <i>END</i>  <i>mise en place des données d'entrée</i>  <i>Copy SUCCESS</i>  <i>END</i></p>	
<p><u>working order</u> : IASI_1_wo_001  <u>IASI L0 data</u> : J1301 ( see § 3.5.2) ( one dump)  <u>Product model</u> : J1301 ( see § 3.5.2) ( 5 files )  <u>AVHRR product</u> : J1 ( see § 3.5.2)  <u>Other and Stable Parameters , Spectral Database</u> : J1301 ( see § 3.5.2)  <u>IASI Context file</u> : J1 ( see § 3.5.2)  <u>Auxiliary data</u> : J1 ( see § 3.5.2)</p>		
<p>4. <i>Send then a STEP command to the OPS subsystem :</i>  <i>Enter STEP command in the PGF simulator window :</i>  <b>STEP 1 input/IASI_1_wo_001</b></p> <p><i>Wait for the processing end.</i>  Stops the OPS</p>	<p>Verify that the STEP command is acknowledged.  ACK STEP 1 0  Verify that WorkingOrder 1 as begun :  ACK START 1 0  Verify the process terminates correctly =&gt; screen message ACK START 1 1</p> <ul style="list-style-type: none"> <li>• Verify after step 10 that</li> <li>• Product files are generated</li> </ul> <p>Report files are generated and notify no error</p>	

working order : IASI\_1\_wo\_002

IASI L0 data : J1 ( see § 3.5.2 ) ( one dump )

Product model : J1 ( see § 3.5.2 ) ( 5 files )

AVHRR product : J1 ( see § 3.5.2 )

Other and Stable Parameters , Spectral Database : J1302 ( see § 3.5.2 )

IASI Context file : J1 ( see § 3.5.2 )

Auxiliary data : J1 ( see § 3.5.2 )

<p>5. <i>Send then a STEP command to the OPS subsystem :</i></p> <p><i>Enter STEP command in the PGF simulator window :</i></p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p><i>Wait for the processing end.</i></p> <p>Stops the OPS</p>	<p>Verify that the STEP command is acknowledged.</p> <p>ACK STEP 1 0</p> <p>Verify that WorkingOrder 1 as begun :</p> <p>ACK START 1 0</p> <p>Verify the process terminates correctly =&gt; screen message ACK START 1 1</p> <ul style="list-style-type: none"> <li>• Verify after step 10 that</li> <li>• Product files are generated</li> </ul> <p>Report files are generated and notify no error</p>	
--	---	--

working order : IASI\_1\_wo\_004

IASI L0 data : J1304 ( see § 3.5.2 ) ( one dump )

Product model : J1 ( see § 3.5.2 ) ( 5 files )

AVHRR product : J1 ( see § 3.5.2 )

Other and Stable Parameters , Spectral Database : J1 ( see § 3.5.2 )

IASI Context file : J1 ( see § 3.5.2 )

Auxiliary data : J1 ( see § 3.5.2 )

<p>6. <i>Send then a STEP command to the OPS subsystem :</i></p> <p><i>Enter STEP command in the PGF simulator window :</i></p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p><i>Wait for the processing end.</i></p> <p>Stops the OPS</p>	<p>Verify that the STEP command is acknowledged.</p> <p>ACK STEP 1 0</p> <p>Verify that WorkingOrder 1 as begun :</p> <p>ACK START 1 0</p> <p>Verify the process terminates correctly =&gt; screen message ACK START 1 1</p> <ul style="list-style-type: none"> <li>• Verify after step 10 that</li> <li>• Product files are generated</li> </ul> <p>Report files are generated and notify no error</p>	
--	---	--

working order : IASI\_1\_wo\_005

IASI L0 data : J1305 ( see § 3.5.2) ( one dump)

Product model : J1 ( see § 3.5.2) ( 5 files )

AVHRR product : J1 ( see § 3.5.2)

Other and Stable Parameters : J1305 ( see § 3.5.2)

Spectral Database : J1

IASI Context file : J1 ( see § 3.5.2)

Auxiliary data : J1 ( see § 3.5.2)

<p>7. <i>Send then a STEP command to the OPS subsystem :</i></p> <p><i>Enter STEP command in the PGF simulator window :</i></p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p><i>Wait for the processing end.</i></p> <p>Stops the OPS</p>	<p>Verify that the STEP command is acknowledged.</p> <p>ACK STEP 1 0</p> <p>Verify that WorkingOrder 1 as begun :</p> <p>ACK START 1 0</p> <p>Verify the process terminates correctly =&gt; screen message ACK START 1 1</p> <ul style="list-style-type: none"> <li>• Verify after step 10 that</li> <li>• Product files are generated</li> </ul> <p>Report files are generated and notify no error</p>	
<p><u>working order</u> : IASI_1_wo_006</p> <p><u>IASI L0 data</u> : J13 ( see § 3.5.2) ( one dump)</p> <p><u>Product model</u> : J1 ( see § 3.5.2) ( 5 files )</p> <p><u>AVHRR product</u> : J1 ( see § 3.5.2)</p> <p><u>Other and Stable Parameters</u> : J1306 ( see § 3.5.2)</p> <p><u>Spectral Database</u> : J1</p> <p><u>IASI Context file</u> : J1 ( see § 3.5.2)</p> <p><u>Auxiliary data</u> : J1 ( see § 3.5.2)</p>	<p>8. <i>Send then a STEP command to the OPS subsystem :</i></p> <p><i>Enter STEP command in the PGF simulator window :</i></p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p><i>Wait for the processing end.</i></p> <p>Stops the OPS</p>	
<p>8. <i>Send then a STEP command to the OPS subsystem :</i></p> <p><i>Enter STEP command in the PGF simulator window :</i></p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p><i>Wait for the processing end.</i></p> <p>Stops the OPS</p>	<p>Verify that the STEP command is acknowledged.</p> <p>ACK STEP 1 0</p> <p>Verify that WorkingOrder 1 as begun :</p> <p>ACK START 1 0</p> <p>Verify the process terminates correctly =&gt; screen message ACK START 1 1</p> <ul style="list-style-type: none"> <li>• Verify after step 10 that</li> <li>• Product files are generated</li> </ul> <p>Report files are generated and notify no error</p>	

<p>7. Dump the content- of the product files. Type Dumpeprod.sh</p>	<p>Each ENG and 1C file is dumped with. a validation tool in \$WORK_DIR/output</p> <p>Dumped files are dumped in subdirectories named GRAN_*</p> <p>Product files corresponding to J130x input data are dumped in GRAN_N1C_x.</p>	
<p>8. Dump the content of CNES reference files Dumperef.sh In order to dump the reference files</p>	<p>Each ENG and 1C file is dumped with. a validation tool in \$WORK_DIR/reference</p> <p>Dumped files are dumped in subdirectories named GRAN_*</p>	
<p>9. Compare the content- of the product files Type : Compare.sh</p>	<p>The previous files are compared with references files using the “compfic” tool. “.diff” files are produced in “\$WORKDIR/output/GRAN_N1***/”</p> <p>The tool must assume no differences for specific data used in algorithms :</p> <ul style="list-style-type: none"> <li>• 23_SSD,</li> <li>• 21_SSS,</li> <li>• 24_IAX,</li> <li>• 43_ISF,</li> <li>• 20_DOC,</li> <li>• 111_MCX,</li> <li>• 32_HEC,</li> <li>• 22_SOS</li> </ul> <p>In accordance for input data sets J130x.</p> <p>For each data type a relative precision is choosen and is used to compare the results.</p> <p>NB : the investigation mode may also be used in this test for more precise results</p>	<p><a href="#">[PPF-0490]</a></p>

## 5.2.49.ALGO-OPS-N-05

Non joué en V1

Test Sheet	ALGO-OPS-N-05	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the IASI L1 numerical results in case of specific data</p> <p>Used data set :J14</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
1. Launch the OPS : Execute the first 3 steps of RUN-OPS-N-01 test.	The OPS is ready to process input data in nominal mode.	
2. Connected as iasi_1 . You must go in the test specific directory : <code>cd \$WORK_DIR/OPS/data/TV/CONF_OPS_N_02</code> then execute the test shell : <code>./lancetestbis.sh</code>		
3. Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)	Verify with the <b>Is \$WORK_DIR/input/aux_data</b> commands that all the input data are available :	
<p><u>AVHRR product</u> : J1 ( see § 3.5.2)</p> <p><u>Other and Stable Parameters , Spectral Database</u> : J1 ( see § 3.5.2)</p> <p><u>IASI Context file</u> : J1 ( see § 3.5.2)</p> <p><u>Auxiliary data</u> : J1 ( see § 3.5.2)</p>		

<p>4. Copy the processing data, needed to process the granule, in the input directory (using symbolic links)</p>	<p>Verify with the</p> <p><b>ls \$WORK_DIR/input</b></p> <p><b>ls \$WORK_DIR/input/product_model</b></p> <p><b>ls \$WORK_DIR/input/unproc_data</b></p> <p>commands that all the input data are available :</p>	
<p><u>working order</u> : IASI_1_wo_001</p> <p><u>IASI L0 data</u> : J14 ( see § 3.5.2) (one granule)</p> <p><u>Product model</u> : J14 ( see § 3.5.2) ( 5 files )</p>		
<p>5. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window :</p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p>Wait for the processing end.</p>	<p>Verify that the STEP command is acknowledged.,</p>	
<p>10. Verify the content- of the product files</p>	<p>Each file is dumped with. A validation tool to verify that the output product are in accordance with the GPFS format. (for exemple , the header and one reccord of each type).</p> <p>Dump some lines of the dump.(AD)</p>	
<p>11. Compare the content- of the product files</p>	<p>The previous files can be compared with references files using the “compfic” tool.</p> <p>The tool must assume no differences for specific data ( MPHR and every lines):</p> <ul style="list-style-type: none"> <li>• (AD)</li> </ul> <p>for each data type a relative precision is choosen and is used to compare the results.</p>	

## 5.2.50.ALGO-OPS-N-06

Test Sheet	ALGO-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing in endurance case</p> <p>Used data set : J1</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p>		
Description :	Expected Result :	Tested Requirement:
1. Replay the PERF-OPS-N-02 test (Execute the first steps)		
2. Verify the content- of the product files.(manual) <b>cd output</b> <b>TraceProduit “ product file name ”</b> <b>GRANx AUTO</b> <b>cd GRANx</b>  <b>cd references</b> <b>TraceProduit “ product file name ”</b> <b>GRANx AUTO</b>	The first granule of the second dump is dumped with. A validation tool. (for exemple , the two first lines of L1C product ).  The reference data is also dumped in the same way	[IF-ALGO-4.4-131] [IF-ALGO-4.4-140] [IF-ALGO-4.4-160] [IF-ALGO-4.4-170] [IF-ALGO-4.4-180] [IF-ALGO-4.4-183]



<p>3. Compare the content- of the product files (automatic)</p> <p><b>cd output</b></p> <p><b>compfic MDRx.tra</b></p> <p><b>.././references/GRANx/MDRx</b></p> <p><b>MDRx.tra.diff 10</b></p> <p><b>.././input/précision.tra</b></p>	<p>The previous files can be compared with references files using the “compfic” tool. For the two first lines</p> <p>Analyse the number of differences ( key word NBDIFFBALEUR) in file <b>MDRx.tra.diff</b>.</p> <p>The tool must assume no differences:</p> <ul style="list-style-type: none"> <li>• spectra,</li> <li>• image,</li> </ul> <p>for each data type a relative precision or an absolute precision is chosen and is used to compare the results (file input/precision.tra).</p>	<p><a href="#">[E_ENC4]</a></p> <p>[PPF-0340]</p> <p>[E_CON4]</p>
---	---	---

## 5.2.51.ALGO-OPS-N-07

Test Sheet	ALGO-OPS-N-07	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing in granule mode</p> <p>Used data set : J0</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p>		
Description :	Expected Result :	Tested Requirement:
1. Replay the GRAN-OPS-N-16 test (Execute the first 5 steps)		
2. <i>Verify the content- of the product files.(automatic) (1C and ENG)</i>  <i>cd output</i>  <i>TraceProduit “product file name ”</i> <i>GRANx AUTO</i>	<p>Each file is dumped with. A validation tool. (, the two first granule of L1C product ).</p> <p>Dump all the lines of the two first granules</p> <p>(NB : the reference files must have been dumped before)</p> <p><b>cd references</b></p> <p><b>dump.ksh</b></p>	<p>[IF-ALGO-4.4-131]</p> <p>[IF-ALGO-4.4-140]</p> <p>[IF-ALGO-4.4-160]</p> <p>[IF-ALGO-4.4-170]</p> <p>[IF-ALGO-4.4-180]</p> <p>[IF-ALGO-4.4-183]</p>

<p>3. <i>Compare the content- of the product files (automatic)</i></p> <p><i>compfic ../references/GRANx/MDRx.tra GRANx/MDRx.tra GRANx/MDRx.tra.diff\$prec ../input/precision.tra</i></p>	<p>The previous files can be compared with references files using the “compfic” tool.</p> <p>Analyse the number of differences ( key word NBDIFFBALEUR) in file <b>MDRx.tra.diff. or VIADRxx.tra.diff</b> in each directory output/GRANx</p> <p>The tool must assume no differences for specific data ( MDR or VIADR):</p> <ul style="list-style-type: none"> <li>• Spectra (MDR) (1C product ), Key word <b>SPECTRE</b></li> <li>• image (MDR)(1C product ), Key word <b>IMAGE</b></li> </ul> <p>(V1 bis only for the following data)</p> <ul style="list-style-type: none"> <li>• spectral shifts and associated quality (MDR) (ENG product ) (SSD), Key words <b>ENG_GSSDWNShift and ENG_GSSDWNShiftQUAL</b></li> <li>• position of the interferometric axis ( before and after filtering) (MDR) (ENG product) (IAX, FAX), Key words <b>ENG_GIAXAXE and ENG_GFAXAXE</b></li> </ul> <p>spectral database interpolation (MDR) (1C product ) (ISF), Key words <b>N1C_GISFLINORIGIN and N1C_GISFCOLORIGIN and N1C_GISFPDS</b> <li>• extremum spectra (VIADR) (ENG product) (EXS), Key word <b>VIADRENG_MEXSS</b></li> <li>• bad pixels survey (VIADR and MDR ) (ENG product ) (DPT), Key words <b>VIADRENG_MDPTIISBADHEALTHPIX and ENG_MDPTVARIMAG</b></li> <li>• radiometric calibration noise (MDR) (ENG product ),(MCX), Key words <b>ENG_MMCTXNOISECALRAD and ENG_MMCTXBIASCALRAD</b></li> <li>• spectra under and overflow (MDR) (ENG product ) (DOC), Key words <b>ENG_GDOCPOSOVERFLOW</b></li> </p>	<p><a href="#">[E_ENC4]</a> [PPF-0340]</p>
---	---	--

(following analysis for V2.1)

- co-localisation IASI-AVHRR (MDR) (1C product) (IAC),

key-words `ENG_GIACOFFSETIISAVHRR` and `ENG_GCCSOFFSETSONDIIS`

- localisation (MDR) (1C product) (GEO),

key-word `N1C_GGEOSONDLOC`

- 41\_CCS output (MDR) (1C product) (CCS),

key-words `N1C_GCCSRADANALX(Y)` and `N1C_GCCSRADANALMEAN(STD)`

for each data type a relative precision is choosen and is used to compare the results(file input/precision.tra).

The data gap caused by a calibration mode on a non earth sight is not computed an the ISRFEM axis is filtered without using these data ( see the context file)

## 5.2.52.ALGO-OPS-N-08

Test Sheet	ALGO-OPS-N-08	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing in granule mode for the scientific validation</p> <p>Used data set : J0V3</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>10 Gbytes of data are available.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/ALGO_OPS_N_08</pre> <p>then execute the shell :</p> <pre>./lancetest.sh</pre> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>		
<p>3. <i>Launch the OPS in standalone mode:</i></p> <pre>..././cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetest.sh):</i></p>	The OPS is ready to process input data in nominal mode.	
<p>4. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i></p>		

AVHRR product : J0V3 ( see § 3.5.2)

Other and Stable Parameters , Spectral Database : J0V3 ( see § 3.5.2)

IASI Context file : J0V3 (see § 3.5.2) this file is copied from the delivery

Auxiliary data : J0V3 ( see § 3.5.2)

5. *Copy the processing data, needed to process all granule, in the input directory (using symbolic links)*

:

working order : IASI\_1\_wo\_001 to IASI\_1\_wo\_041

IASI L0 data : J0V3 ( see § 3.5.2) ( all granule)

Product model : J0V3 ( see § 3.5.2) (all granule)

6. Send then a STEP command to the OPS subsystem :

*Enter STEP command in the PGF simulator window for all the granules (41) :*

**STEP 1 input/IASI\_1\_wo\_001**

.....

*Wait for the processing end.*

Verify that each STEP command is acknowledged.

(file CMD\_xxxx)

Verify that with the last granule the processing chain stops

(file CMD\_xxxx)

Verify that the predicted files are produced in the output directory :

- IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated
- Report file :

<p>7. Dump the content- of the product files (1A/1B/1C and ENG) for three granules and compare with references</p> <p><b>(orbit beginning)</b></p> <p><b>diff.sh xx 1 DETAIL</b></p> <p><b>(pole)</b></p> <p><b>diff.sh xx 14 DETAIL</b></p> <p><b>(orbit middle)</b></p> <p><b>diff.sh xx 23 DETAIL</b></p> <p><b>(orbit end)</b></p> <p><b>diff.sh xx 41 DETAIL</b></p> <p>(NB : xx = 1A / 1B / 1C / ENG according the product type)</p> <p>(NB2 : 1A and 1B granules are compared only for granule 14)</p>	<p>Each product is dumped (1A/1B/1C/ENG) and compared with reference data. Then for a list of fields précised in the files input/keylist_xx.txt (xx=1A, 1B, 1C, ENG) the difference is analysed and summarized in file output/diff_xx.txt (xx=1A, 1B, 1C, ENG)</p> <p>(NB 1: the reference files must have been dumped before using the command</p> <p><b>dumpref.ksh 1</b></p> <p><b>dumpref.ksh 14</b></p> <p><b>dumpref.ksh 23</b></p> <p><b>dumpref.ksh 41 )</b></p> <p>(NB 2: the precision used to compare the results with the reference is described in several tables here under</p>	
<p>8. <i>Analyse the differences</i></p> <p><b>cd output</b></p> <p><b>more diff_1A_x.txt</b></p> <p><b>more diff_1B_x.txt</b></p> <p><b>more diff_1C_x.txt</b></p> <p><b>more diff_ENG_x.txt</b></p> <p>(NB : xx = 1, 14,23, 41 according the granule number)</p> <p>(NB2 : 1A and 1B granules are compared only for granule 14)</p> <p>(NB3 : CCS results are compared only for granule 23)</p>	<p>For each file the content must show that there is no differences using the current precision (define in file input/precision.tra )</p> <p>The file must contain several records of the following type (x is variable) :</p> <p>-- debut du fichier:output/GRAN_XXX_x/MDRx.tra.diff --</p> <p>-- mot cle recherché xxxxxxxxxxxxxxxx --</p> <p>xxxxxxxxxxxxxxxx NB DIFF <b>NO_DIFF</b></p> <p>-- fin du fichier:output/GRAN_ENG_1/MDR1.tra.diff --</p> <p>....</p> <p><b>NB : at the beginning of the file a summary is done</b></p>	

<p>9. Analyse the differences in non automatic case (MPHR, GIADR)</p> <p><b>cd output</b></p> <p><b>TraceProduit “product file name ”</b></p> <p><b>GRANyyxxbis</b></p> <p>(NB : xx = 1, according the granule number)</p> <p>(NB : yy = 1C / ENG according the product type)</p> <p>and choose the option to dump GIADR and MPHR</p> <p>NB : you must do the same for the reference products</p>	<p>For each file, the content must show that there is no differences with the reference data.</p> <p><b>cd GRANyyxxbis</b></p> <p>more GIADR1.tra</p> <p>more GIADR2.tra</p> <p>more MPHR.tra</p> <p>Compare the keywords précised in the tables hereunder.</p>	
---	---	--

Record type	Field / precision	
MDR 1A	DEGRADED_INST_MDR	égalité
	DEGRADED_PROC_MDR	égalité
	GEPSIasiMode	égalité
	GEPSOPSProcessingMode	égalité
	GEPSIdConf	égalité
	GEPSLocIasiAvhrr_IASI	1/60 secondes en ligne
	GEPSLocIasiAvhrr_IIS	et 1/10 pixel en colonne
	GlsfLinOrigin	égalité
	GlsfColOrigin	égalité
	GlsfPds1	1
	GlsfPds2	1
	GlsfPds3	1



Record type	Field / precision	
	GlsfPds4	1
	GlrclImage	0.02 K
	GQisFlagQual	égalité
	GQisQualIndex	0,01
	GQisQualIndexIIS	0,01
	GQisQualIndexLoc	0,1
	GQisQualIndexRad	0,01
	GQisQualIndexSpect	0,01
	GQisSysTecIISQual	égalité
	GQisSysTecSondQual	égalité
	GGeoSondLoc	100m
	GGeoSondAnglesMETOP	0,01°
	GGeolISAnglesMETOP	0,01°
	GGeoSondAnglesSUN	0,1°
	GGeolISAnglesSUN	0,1°
	GGeolISLoc	100m
	EARTH_SATELLITE_DISTANCE	100m
	IDefSpectrDWn	égalité
	IDefNsfirst	égalité
	IDefNslast	égalité
	GSmcSpect	0,03 K
	IDefCovarMatEigenVal1	égalité

Record type	Field / precision	
MDR 1B	DEGRADED_INST_MDR	égalité
	DEGRADED_PROC_MDR	égalité
	GEPSIasiMode	égalité
	GEPSOPSProcessingMode	égalité
	GEPSIdConf	égalité
	GEPSLocIasiAvhrr_IASI	1/60 secondes en ligne
	GEPSLocIasiAvhrr_IIS	et 1/10 pixel en colonne
	GlsfLinOrigin	égalité
	GlsfColOrigin	égalité
	GlsfPds1	1
	GlsfPds2	1
	GlsfPds3	1
	GlsfPds4	1
	GlrclImage	0.02 K
	GQisFlagQual	égalité
	GQisQualIndex	0,01
	GQisQualIndexIIS	0,01
	GQisQualIndexLoc	0,1
	GQisQualIndexRad	0,01
	GQisQualIndexSpect	0,01
	GQisSysTecIISQual	égalité
	GQisSysTecSondQual	égalité
	GGeoSondLoc	100m
	GGeoSondAnglesMETOP	0,01°

Record type	Field / precision	
	GGeolISAnglesMETOP	0,01°
	GGeoSondAnglesSUN	0,1°
	GGeolISAnglesSUN	0,1°
	GGeolISLoc	100m
	EARTH_SATELLITE_DISTANCE	100m
	IDefSpecDwn1b	égalité
	IDefNsfirst1b	égalité
	IDefNslast1b	égalité
	GS1bSpect	0,03 K
	IDefCovarMatEigenVal1b	égalité

Record type	Field / precision	
MDR 1C	DEGRADED_INST_MDR	égalité
	DEGRADED_PROC_MDR	égalité
	GEPSIasiMode	égalité
	GEPSOPSProcessingMode	égalité
	GEPSIdConf	égalité
	GEPSLocIasiAvhrr_IASI	1/60 secondes en ligne
	GEPSLocIasiAvhrr_IIS	et 1/10 pixel en colonne
	GlsfLinOrigin	égalité
	GlsfColOrigin	égalité
	GlsfPds1	1
	GlsfPds2	1
	GlsfPds3	1

Record type	Field / precision	
	GlsfPds4	1
	Glrclmage	0.02 K
	GQisFlagQual	égalité
	GQisQualIndex	0,01
	GQisQualIndexIIS	0,01
	GQisQualIndexLoc	0,1
	GQisQualIndexRad	0,01
	GQisQualIndexSpect	0,01
	GQisSysTecIISQual	égalité
	GQisSysTecSondQual	égalité
	GGeoSondLoc	100m
	GGeoSondAnglesMETOP	0,01°
	GGeoIISAnglesMETOP	0,01°
	GGeoSondAnglesSUN	0,1°
	GGeoIISAnglesSUN	0,1°
	GGeoIISLoc	100m
	EARTH_SATELLITE_DISTANCE	100m
	GS1cSpect	0,03 K
	IDefCovarMatEigenVal1c	égalité
	IDefCcsChannelId	égalité
	GCcsRadAnaINbClass	Mettre 0 pour avoir la statistique
	B0	B0 écart max 10-6

Record type	Field / precision	
	B1	B1 écart max <0.2K sur canaux 2 et 3 B1 écart max < 0.1 sur canaux 1, 2 et 3
	B2	B2 10-7 à 1 sigma et écart max à 10-6

Record type	Field / precision	
MDR ENG	BIMSBTT	égalité
	GFtbFilterdBBT	0,001K
	GCcsConfAvhrrChannel	égalité
	CGeoSubSatellitePosition	100 m
	GSmeTScan	0,1K
	GSmeFlagDateNok	égalité
	GFtBBBTRes	0,002 K
	GFtBFlagBBTNonQual	égalité
	GDocFlagUnderOverflow	égalité
	GDocNbUnderFlow	égalité
	GDocNbOverflow	égalité
	GDocPosUnderFlow	égalité
	BCodSpecVeriff	égalité
	GlacOffsetIIISAvhrr	égalité
		0,1 pixel à 1 sigma

Record type	Field / precision	
	GlacCorrelQual	10-3
	GlacPosMaxQual	10-2
	GlacFlagCoregNonValid	égalité
	GlacFlagCoregNonQual	égalité
	GlacVarImaglIS	1%
	GCcsOffsetSondAvhrr	0,1 pixel à 1 sigma
	GCcsOffsetSondIIS	5,10-4 degrés
	GCcsQualIndex	non utilisé
	GCcsFlagDateNOK	égalité
	CGCcsRadAnalNbClass	égalité
	GCcsFlagPostProcessing	égalité
	GCcsNonClassifRate	0,01
	GCcsVarianceRate	0,01
	GSsdWnShift	2.35e-3 m-1 (~1.10-8)
	GSsdWnShiftQual	0,1
	GSsdFlagSpectralShiftNonQual	égalité
	GSssWnShiftMean	2.35e-3 m-1 (~1.10-8)
	GSssWnShiftMeanQual	1 e-3
	GSssFlagNonSelPix	égalité
	GSssFlagDateNOK	égalité
	GlaxAxeY	1.10-6 rad
	GlaxAxeZ	1.10-6 rad
	GlaxFlagAxeNonQual	égalité

Record type	Field / precision	
	GlaxAxeQual	5e-6 rad
	GlaxAxeRes	1,0e-06 m-2
	GFaxAxeY	1.10-6 rad
	GFaxAxeZ	1.10-6 rad
	GFaxFlagAxeNonQual	égalité
	GFaxAxeRes	1,0e-06 m-2
	GlsfFlagPdsNonValid	égalité
	GQisFlagQual	égalité
	GQisQualIndex	0,01
	GQisQualIndexIIS	0,01
	GQisQualIndexLoc	0,1
	GQisQualIndexRad	0,01
	GQisQualIndexSpect	0,01
	MMcxNoiseCalRad	0.01 K Nedt
	MMcxBiasCalRad	0.01 K Nedt
	MMcxFlagNoiseCalRad	égalité
	MMcxFlagBiasCalRad	égalité
	MMcxCoeffCalRad	0.001
	GHecFlagDateNOK	égalité
	Data_PX	égalité
	Data_IP	égalité

Product type	Field / precision	
VIADR ENG	MEXSMIN	1 pas de codage
	MEXSMAX	1 pas de codage

Record type	Field / precision	
GIADR_SCALEFACTOR	IdefScaleSondNbScale	égalité
	IdefScaleSondNsFirst	égalité
	IdefScaleSondNsLast	égalité
	IdefScaleSondScaleFactor	égalité
	IdefScaleIISScaleFactor	égalité

Record type	Field / precision	
GIADR_QUALITY	IDefPsfSondOverSampFactor	égalité
	IDefPsfSondY	égalité
	IDefPsfSondZ	égalité
	IDefPsfSondWgt	égalité
	IDefIISSrfNsfirst	égalité
	IDefIISSrfNsIast	égalité
	IDefIISSrf	égalité
	IDefIISSrfDWn	égalité
	IDefIISNeDT	égalité
	IDefDptIISSDeadPix	égalité

record type	Field / precision	
MPHR	PARENT_PRODUCT_NAME_1	égalité



record type	Field / precision	
	PARENT_PRODUCT_NAME_2	égalité
	PARENT_PRODUCT_NAME_3	égalité
	PARENT_PRODUCT_NAME_4	égalité
	SENSING_START	égalité
	SENSING_END	égalité
	SENSING_START	égalité
	SENSING_END	égalité
	PROCESSING_CENTRE	égalité
	PROCESSOR_MAJOR_VERSION	égalité
	SUBSAT_LATITUDE_START	égalité
	SUBSAT_LONGITUDE_START	égalité
	SUBSAT_LATITUDE_END	égalité
	SUBSAT_LONGITUDE_END	égalité
	COUNT_DEGRADED_INST_MDR	égalité
	COUNT_DEGRADED_PROC_MDR	égalité
	COUNT_DEGRADED_INST_MDR_BLOCKS	égalité
	COUNT_DEGRADED_PROC_MDR_BLOCKS	égalité
	DURATION OF PRODUCT	égalité
	MILLISECONDS_OF_DATA_PRESENT	égalité
	MILLISECONDS_OF_DATA DISSING	égalité

### 5.2.53.ALGO-OPS-N-09

Test Sheet	ALGO-OPS-N-09	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing in granule mode for the scientific validation (Image calibration and geolocalisation )</p> <p>Used data set : J1V3 (imagery)</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>250 Mbytes of data are available.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/ALGO_OPS_N_09</code></p> <p>then execute the shell :</p> <p><code>./lancetest.sh</code></p> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <p><code>cd \$WORK_DIR/OPS/tools/OPStools</code></p> <p><code>Clean.sh</code></p>		
<p>3. <i>Launch the OPS in standalone mode:</i></p> <p><code>./.././../cmd/StartStandAlone.sh 1</code>  <code>\$WORK_DIR N (this step is included in the shell lancetest.sh):</code></p>	The OPS is ready to process input data in nominal mode.	
<p>4. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i></p>		

AVHRR product : J0V2 ( see § 3.5.2)

Other and Stable Parameters , Spectral Database : J1V3 ( see § 3.5.2)

IASI Context file : J0V2 (see § 3.5.2) this file is copied from the delivery

Auxiliary data : J0V2 ( see § 3.5.2)

5. *Copy the processing data, needed to process all granule, in the input directory (using symbolic links)*

:

working order : IASI\_1\_wo\_001

IASI L0 data : J1V3 ( see § 3.5.2) ( all granule)

Product model : J0V2 ( see § 3.5.2) (all granule)

6. Send then a STEP command to the OPS subsystem :

*Enter STEP command in the PGF simulator window for the granule :*

**STEP 1 input/IASI\_1\_wo\_001**

.....

*Wait for the processing end.*

Verify that each STEP command is acknowledged.

(file CMD\_xxxx)

Verify that with the last granule the processing chain stops

(file CMD\_xxxx)

Verify that the predicted files are produced in the output directory :

- IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated
- Report file :

7. Dump the content- of the product files (1A/1B/1C and ENG) and compare with references

**diff.sh xx 1 DETAIL**

(NB : xx = 1A / 1B / 1C / ENG according the product type)

Each product is dumped (1A/1B/1C/ENG) and compared with reference data. Then for a list of fields précised in the files input/keylist\_xx.txt (xx=1A, 1B, 1C, ENG) the difference is analysed and summarized in file output/diff\_xx.txt (xx=1A, 1B, 1C, ENG)

(NB 1: the reference files must have been dumped before using the command

**dumpref.ksh 1)**

(NB 2: the precision used to compare the results with the reference is described in several tables here under )

<p>8. <i>Analyse the differences</i></p> <p><i>cd output</i></p> <p><i>more diff_1A_1.txt</i></p> <p><i>more diff_1B_1.txt</i></p> <p><i>more diff_1C_1.txt</i></p> <p><i>more diff_ENG_1.txt</i></p> <p>(NB3 : CCS results are not compared)</p>	<p>For each file the content must show that there is no differences using the current precision ( define in file input/precision.tra)</p> <p>The file must contain several records of the following type (x is variable) :</p> <p>-- debut du fichier:output/GRAN_XXX_x/MDRx.tra.diff --</p> <p>-- mot cle recherché xxxxxxxxxxxxxxxx --</p> <p>xxxxxxxxxxxxxxxx NB DIFF <b>NO_DIFF</b></p> <p>-- fin du fichier:output/GRAN_ENG_1/MDR1.tra.diff --</p> <p>....</p> <p><b>NB : at the beginning of the file a summary is done</b></p>	
---	---	--

Record type	Field / precision	
MDR 1A	DEGRADED_INST_MDR	égalité
	DEGRADED_PROC_MDR	égalité
	GlrcImage	0.02 K
	GQisFlagQual	égalité
	GQisQualIndex	0,01
	GQisQualIndexIIS	0,01
	GQisQualIndexLoc	0,1
	GQisQualIndexRad	0,01
	GQisQualIndexSpect	0,01
	GQisSysTecIISQual	égalité
	GQisSysTecSondQual	égalité
	GGeoSondLoc	100m
	GGeoSondAnglesMETOP	0,01°
	GGeoIISAnglesMETOP	0,01°
	GGeoSondAnglesSUN	0,1°

Record type	Field / precision	
	GGeolISAnglesSUN	0,1°
	GGeolISLoc	100m
	GSmcSpect	0,03 K

Record type	Field / precision	
MDR 1B	DEGRADED_INST_MDR	égalité
	DEGRADED_PROC_MDR	égalité
	Glrclmage	0.02 K
	GQisFlagQual	égalité
	GQisQualIndex	0,01
	GQisQualIndexIIS	0,01
	GQisQualIndexLoc	0,1
	GQisQualIndexRad	0,01
	GQisQualIndexSpect	0,01
	GQisSysTecIISQual	égalité
	GQisSysTecSondQual	égalité
	GGeoSondLoc	100m
	GGeoSondAnglesMETOP	0,01°
	GGeolISAnglesMETOP	0,01°
	GGeoSondAnglesSUN	0,1°
	GGeolISAnglesSUN	0,1°
	GGeolISLoc	100m

Record type	Field / precision	
MDR 1C	DEGRADED_INST_MDR	égalité
	DEGRADED_PROC_MDR	égalité
	GlrclImage	0.02 K
	GQisFlagQual	égalité
	GQisQualIndex	0,01
	GQisQualIndexIIS	0,01
	GQisQualIndexLoc	0,1
	GQisQualIndexRad	0,01
	GQisQualIndexSpect	0,01
	GQisSysTecIISQual	égalité
	GQisSysTecSondQual	égalité
	GGeoSondLoc	100m
	GGeoSondAnglesMETOP	0,01°
	GGeoIISAnglesMETOP	0,01°
	GGeoSondAnglesSUN	0,1°
	GGeoIISAnglesSUN	0,1°
	GGeoIISLoc	100m
	IDefCcsChannelId	égalité
	GCcsRadAnaINbClass	Mettre 0 pour avoir la statistique
	B0	B0 écart max 10-6

Record type	Field / precision	
	B1	B1 écart max <0.2K sur canaux 2 et 3 B1 écart max < 0.1 sur canaux 1, 2 et 3
	B2	B2 10-7 à 1 sigma et écart max à 10-6

Record type	Field / precision	
MDR ENG	GEPSGranulNumber	égalité
	GEPSEndEclipseTime	1 seconde
	GEPS_LN	égalité
	BCodSpecVeriff	égalité
	GlacOffsetIISAvhrr	0,1 pixel à 1 sigma
	GlacCorrelQual	10-3
	GlacPosMaxQual	10-2
	GlacFlagCoregNonValid	égalité
	GlacFlagCoregNonQual	égalité
	GlacVarImaglIS	1%
	GCcsOffsetSondAvhrr	0,1 pixel à 1 sigma
	GCcsQualIndex	non utilisé
	GCcsFlagDateNOK	égalité
	CGCcsRadAnalNbClass	égalité
	GCcsFlagPostProcessing	égalité
	GCcsNonClassifRate	0,01

Record type	Field / precision	
	GCcsVarianceRate	0,01
	GlccRadCalOffsetImag	égalité
	GlccRadCalSlopeImag	égalité
	GlccFlagInit	égalité
	GQisFlagQual	égalité
	GQisQualIndex	0,01
	GQisQualIndexIIS	0,01
	GQisQualIndexLoc	0,1
	GQisQualIndexRad	0,01
	GQisQualIndexSpect	0,01
	MDptVarImagMax	1e-4 K2
	MDptVarImagMean	1e-4 K2
	MDptPixQual	0,0002
	Data_PX	égalité
	Data_IP	égalité

Product type	Field / precision	
VIADR ENG	MEXSMIN	1 pas de codage
	MEXSMAX	1 pas de codage
	MDptIISBadHealthPix	égalité
	MDptIISinHomPix	égalité



## 5.2.54.ALGO-OPS-N-10

Test Sheet	ALGO-OPS-N-10	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing in granule mode for the scientific validation (degraded case for CCS 41)</p> <p>Used data set : J2V2</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test. 250 Mbytes of data are available.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/ALGO_OPS_N_10</pre> <p>then execute the shell :</p> <pre>./lancetest.sh</pre> <p><b>This shell execute automaticaly the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>		
<p>3. <i>Launch the OPS in standalone mode:</i></p> <pre>./.././../cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetest.sh):</i></p>	The OPS is ready to process input data in nominal mode.	
<p>4. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i></p>		

<u>AVHRR product</u> : none <u>Other and Stable Parameters , Spectral Database</u> : J2V2 ( see § 3.5.2) <u>IASI Context file</u> : J0V2 (see § 3.5.2) this file is copied from the delivery <u>Auxiliary data</u> : J0V2 ( see § 3.5.2)		
5. <i>Copy the processing data, needed to process all granule, in the input directory (using symbolic links)</i>	:	
<u>working order</u> : IASI_1_wo_001 <u>IASI L0 data</u> : J2V2 ( see § 3.5.2) ( all granule) <u>Product model</u> : J0V2 ( see § 3.5.2) (all granule)		
6. Send then a STEP command to the OPS subsystem : <i>Enter STEP command in the PGF simulator window for the granule :</i> <b>STEP 1 input/IASI_1_wo_001</b> ..... <i>Wait for the processing end.</i>	Verify that each STEP command is acknowledged. (file CMD_xxxx) Verify that with the last granule the processing chain stops (file CMD_xxxx) Verify that the predicted files are produced in the output directory : <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul>	
7. Dump the content- of the product files (1A/1B/1C and ENG) and compare with references <b>diff.sh xx 1 DETAIL</b>  (NB : xx = 1A / 1B / 1C / ENG according the product type)	Each product is dumped (1A/1B/1C/ENG) and compared with reference data. Then for a list of fields précised in the files input/keylist_xx.txt (xx=1A, 1B, 1C, ENG) the difference is analysed and summarized in file output/diff_xx.txt (xx=1A, 1B, 1C, ENG)  (NB 1: the reference files must have been dumped before using the command <b>dumpref.ksh 1)</b> (NB 2: the precision used to compare the results with the reference is described in several tables here under)	

<p>8. <i>Analyse the differences</i></p> <p><i>cd output</i></p> <p><i>more diff_1A_1.txt</i></p> <p><i>more diff_1B_1.txt</i></p> <p><i>more diff_1C_1.txt</i></p> <p><i>more diff_ENG_1.txt</i></p>	<p>For each file the content must show that there is no differences using the current precision ( define in file input/precision.tra)</p> <p>The file must contain several records of the following type (x is variable) :</p> <pre>-- debut du fichier:output/GRAN_xxx_x/MDRx.tra.diff -- -- mot cle recherché xxxxxxxxxxxxxxxx --       xxxxxxxxxxxxxxxx NB DIFF <b>NO_DIFF</b> -- fin du fichier:output/GRAN_ENG_1/MDR1.tra.diff -- ....</pre>	
---	---	--

Product type	Field / precision
MDR 1A	

Product type	Field / precision
MDR 1B	

Product type	Field / precision
MDR 1C	

Product type	Field / precision
MDR ENG	

Product type	Field / precision
VIADR_ENG	

## 5.2.55.ALGO-OPS-N-11

Test Sheet	ALGO-OPS-N-11	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing in granule mode for the scientific validation (degraded case for CCS 41)</p> <p>Used data set : J3V2</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>250 Mbytes of data are available.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/ALGO_OPS_N_11</pre> <p>then execute the shell :</p> <pre>./lancetest.sh</pre> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>		
<p>3. <i>Launch the OPS in standalone mode:</i></p> <pre>..././cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetest.sh):</i></p>	The OPS is ready to process input data in nominal mode.	
<p>4. <i>Copy at first, the input auxiliary data, needed to process all granules in the input directory (using symbolic links)</i></p>		

AVHRR product : J0V2 ( see § 3.5.2)

Other and Stable Parameters , Spectral Database : J3V2 ( see § 3.5.2)

IASI Context file : J0V2 (see § 3.5.2) this file is copied from the delivery

Auxiliary data : J0V2 ( see § 3.5.2)

5. *Copy the processing data, needed to process all granule, in the input directory (using symbolic links)*

:

working order : IASI\_1\_wo\_001

IASI L0 data : J3V2 ( see § 3.5.2)

Product model : J0V2 ( see § 3.5.2) (all granule)

6. Send then a STEP command to the OPS subsystem :

*Enter STEP command in the PGF simulator window for the granule :*

**STEP 1 input/IASI\_1\_wo\_001**

.....

*Wait for the processing end.*

Verify that each STEP command is acknowledged.

(file CMD\_xxxx)

Verify that with the last granule the processing chain stops

(file CMD\_xxxx)

Verify that the predicted files are produced in the output directory :

- IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated
- Report file :

7. Dump the content- of the product files (1A/1B/1C and ENG) and compare with references

**diff.sh xx 1 DETAIL**

(NB : xx = 1A / 1B / 1C / ENG according the product type)

Each product is dumped (1A/1B/1C/ENG) and compared with reference data. Then for a list of fields précised in the files input/keylist\_xx.txt (xx=1A, 1B, 1C, ENG) the difference is analysed and summarized in file output/diff\_xx.txt (xx=1A, 1B, 1C, ENG)

(NB 1: the reference files must have been dumped before using the command

**dumpref.ksh 1)**

(NB 2: the precision used to compare the results with the reference is described in several tables here under)

<p>8. <i>Analyse the differences</i></p> <p><i>cd output</i></p> <p><i>more diff_1A_1.txt</i></p> <p><i>more diff_1B_1.txt</i></p> <p><i>more diff_1C_1.txt</i></p> <p><i>more diff_ENG_1.txt</i></p>	<p>For each file the content must show that there is no differences using the current precision ( define in file input/precision.tra)</p> <p>The file must contain several records of the following type (x is variable) :</p> <pre>-- debut du fichier:output/GRAN_xxx_x/MDRx.tra.diff -- -- mot cle recherché xxxxxxxxxxxxxxxx --       xxxxxxxxxxxxxxxx NB DIFF <b>NO_DIFF</b> -- fin du fichier:output/GRAN_ENG_1/MDR1.tra.diff -- ....</pre>	
---	---	--

Product type	Field / precision
MDR 1A	

Product type	Field / precision
MDR 1B	

Product type	Field / precision
MDR 1C	

Product type	Field / precision
MDR ENG	

Product type	Field / precision
VIADR_ENG	

**OCNES**

**IASI**

**IA-PE-2100-9559-THA**

Edit. : **03**

Date : **21/02/2003**

Rév. : **08**

Date : **28/07/2009**

Référence : -

Page : 225

## 5.2.56.ALGO-OPS-N-12

Test Sheet	ALGO-OPS-N-12	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing in granule mode for the scientific validation (transition case for CCS 41)</p> <p>Used data set : J4V2</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>250 Mbytes of data are available.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/ALGO_OPS_N_12</pre> <p>then execute the shell :</p> <pre>./lancetest.sh</pre> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>		
<p>3. <i>Launch the OPS in standalone mode:</i></p> <pre>./.././../cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetest.sh):</i></p>	The OPS is ready to process input data in nominal mode.	
<p>4. <i>Copy at first, the input auxiliary data, needed to process all granules in the input directory (using symbolic links)</i></p>		



AVHRR product : J4V2 ( see § 3.5.2)

Other and Stable Parameters , Spectral Database : J0V2 ( see § 3.5.2)

IASI Context file : J0V2 (see § 3.5.2) this file is copied from the delivery

Auxiliary data : J0V2 ( see § 3.5.2)

5. *Copy the processing data, needed to process all granule, in the input directory (using symbolic links)*

:

working order : IASI\_1\_wo\_001

IASI L0 data : J4V2 ( see § 3.5.2)

Product model : J0V2 ( see § 3.5.2) (all granule)

6. Send then a STEP command to the OPS subsystem :

*Enter STEP command in the PGF simulator window for the granule :*

**STEP 1 input/IASI\_1\_wo\_001**

.....

*Wait for the processing end.*

Verify that each STEP command is acknowledged.

(file CMD\_XXXX)

Verify that with the last granule the processing chain stops

(file CMD\_XXXX)

Verify that the predicted files are produced in the output directory :

- IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated
- Report file :

7. Dump the content- of the product files (1A/1B/1C and ENG) and compare with references

**diff.sh xx 1 DETAIL**

(NB : xx = 1C / ENG according the product type)

Each product is dumped (1A/1B/1C/ENG) and compared with reference data. Then for a list of fields précised in the files input/keylist\_xx.txt (xx=1A, 1B, 1C, ENG) the difference is analysed and summarized in file output/diff\_xx.txt (xx=1A, 1B, 1C, ENG)

(NB 1: the reference files must have been dumped before using the command

**dumpref.ksh 1)**

(NB 2: the precision used to compare the results with the reference is described in several tables here under)

<p>8. <i>Analyse the differences</i></p> <p><i>cd output</i></p> <p><i>more diff_1C_1.txt</i></p> <p><i>more diff_ENG_1.txt</i></p>	<p>For each file the content must show that there is no differences using the current precision ( define in file input/precision.tra)</p> <p>The file must contain several records of the following type (x is variable) :</p> <pre>-- debut du fichier:output/GRAN_xxx_x/MDRx.tra.diff -- -- mot cle recherché xxxxxxxxxxxxxxxx --       xxxxxxxxxxxxxxxx NB DIFF <b>NO_DIFF</b> -- fin du fichier:output/GRAN_ENG_1/MDR1.tra.diff -- ....</pre>	
---	---	--

Product type	Field / precision	
MDR 1C	IDefCcsChannelId	égalité
	GCcsRadAnalNbClass	Mettre 0 pour avoir la statistique
	GCcsRadAnalWgt	Statistique sur B0, B1, B2 et B3
	GCcsRadAnalY	
	GCcsRadAnalZ	
	GCcsRadAnalMean	
	GCcsRadAnalStd	Non testé en recette

Product type	Field / precision
--------------	-------------------

Product type	Field / precision	
MDR ENG	GCcsOffsetSondAvhrr	0,1 pixel à 1 sigma
	GCcsQualIndex	non utilisé
	GCcsFlagDateNOK	égalité
	CGCcsRadAnalNbClass	égalité
	CGCcsFlagPostProcessing	égalité
	CGCcsNonClassifRate	0,01
	CGCcsVarianceRate	0,01 mais n'a de sens que si le nb de classe est le même

## 5.2.57.ALGO-OPS-N-13

Test Sheet	ALGO-OPS-N-13	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing in granule mode for the scientific validation (spectrometry post-calibration and spectral calibration)</p> <p>Used data set : J5V3</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>250 Mbytes of data are available.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/ALGO_OPS_N_13</pre> <p>then execute the shell :</p> <pre>./lancetest.sh</pre> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>		
<p>3. <i>Launch the OPS in standalone mode:</i></p> <pre>./.././../cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetest.sh):</i></p>	The OPS is ready to process input data in nominal mode.	
<p>4. <i>Copy at first, the input auxiliary data, needed to process all granules in the input directory (using symbolic links)</i></p>		

<u>AVHRR product</u> : J0V2 ( see § 3.5.2) <u>Other and Stable Parameters , Spectral Database</u> : J5V3 ( see § 3.5.2) <u>IASI Context file</u> : J0V2 (see § 3.5.2) this file is copied from the delivery <u>Auxiliary data</u> : J0V2 ( see § 3.5.2)		
5. <i>Copy the processing data, needed to process all granule, in the input directory (using symbolic links)</i>	:	
<u>working order</u> : IASI_1_wo_001 <u>IASI L0 data</u> : J5V3 ( see § 3.5.2) <u>Product model</u> : J0V2 ( see § 3.5.2) (all granule)		
6. Send then a STEP command to the OPS subsystem :  <i>Enter STEP command in the PGF simulator window for the granule :</i>  <b>STEP 1 input/IASI_1_wo_001</b>  .....  <i>Wait for the processing end.</i>	Verify that each STEP command is acknowledged. (file CMD_xxxx)  Verify that with the last granule the processing chain stops (file CMD_xxxx)  Verify that the predicted files are produced in the output directory : <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul>	
7. Dump the content- of the product files (1A/1B/1C and ENG) and compare with references  <b>diff.sh xx 1 DETAIL</b>  (NB : xx = 1A / 1B / 1C / ENG according the product type)	Each product is dumped (1A/1B/1C/ENG) and compared with reference data. Then for a list of fields précised in the files input/keylist_xx.txt (xx=1A, 1B, 1C, ENG) the difference is analysed and summarized in file output/diff_xx.txt (xx=1A, 1B, 1C, ENG)  (NB 1: the reference files must have been dumped before using the command <b>dumpref.ksh 1)</b> (NB 2: the precision used to compare the results with the reference is described in several tables here under)	

<p>8. <i>Analyse the differences</i></p> <p><i>cd output</i></p> <p><i>more diff_1A_1.txt</i></p> <p><i>more diff_1B_1.txt</i></p> <p><i>more diff_1C_1.txt</i></p> <p><i>more diff_ENG_1.txt</i></p>	<p>For each file the content must show that there is no differences using the current precision ( define in file input/precision.tra)</p> <p>The file must contain several records of the following type (x is variable) :</p> <p>-- debut du fichier:output/GRAN_xxx_x/MDRx.tra.diff --</p> <p>-- mot cle recherché xxxxxxxxxxxxxxxx --</p> <p>xxxxxxxxxxxxxxxx NB DIFF <b>NO_DIFF</b></p> <p>-- fin du fichier:output/GRAN_ENG_1/MDR1.tra.diff --</p> <p>....</p> <p><b>NB : at the beginning of the file a summary is done</b></p>	
---	---	--

## 5.2.58.ALGO-OPS-N-14

Test Sheet	ALGO-OPS-N-14	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing in granule mode for the scientific validation (spectrometry post-calibration and spectral calibration)</p> <p>Used data set : J6V2</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test. 250 Mbytes of data are available.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/ALGO_OPS_N_14</pre> <p>then execute the shell :</p> <pre>./lancetest.sh</pre> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>		
<p>3. <i>Launch the OPS in standalone mode:</i></p> <pre>./.././../cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetest.sh):</i></p>	The OPS is ready to process input data in nominal mode.	
<p>4. <i>Copy at first, the input auxiliary data, needed to process all granules in the input directory (using symbolic links)</i></p>		

AVHRR product : J0V2 ( see § 3.5.2)

Other and Stable Parameters , Spectral Database : J6V2 ( see § 3.5.2)

IASI Context file : J0V2 (see § 3.5.2) this file is copied from the delivery

Auxiliary data : J0V2 ( see § 3.5.2)

5. *Copy the processing data, needed to process all granule, in the input directory (using symbolic links)*

:

working order : IASI\_1\_wo\_001

IASI L0 data : J6V2 ( see § 3.5.2)

Product model : J0V2 ( see § 3.5.2) (all granule)

6. Send then a STEP command to the OPS subsystem :

*Enter STEP command in the PGF simulator window for the granule :*

**STEP 1 input/IASI\_1\_wo\_001**

.....

*Wait for the processing end.*

Verify that each STEP command is acknowledged.

(file CMD\_xxxx)

Verify that with the last granule the processing chain stops

(file CMD\_xxxx)

Verify that the predicted files are produced in the output directory :

- IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated
- Report file :

7. Dump the content- of the product files (1A/1B/1C and ENG) and compare with references

**diff.sh xx 1 DETAIL**

(NB : xx = 1A / 1B / 1C / ENG according the product type)

Each product is dumped (1A/1B/1C/ENG) and compared with reference data. Then for a list of fields précised in the files input/keylist\_xx.txt (xx=1A, 1B, 1C, ENG) the difference is analysed and summarized in file output/diff\_xx.txt (xx=1A, 1B, 1C, ENG)

(NB 1: the reference files must have been dumped before using the command

**dumpref.ksh 1)**

(NB 2: the precision used to compare the results with the reference is described in several tables here under)



<p>8. <i>Analyse the differences</i></p> <p><i>cd output</i></p> <p><i>more diff_1A_1.txt</i></p> <p><i>more diff_1B_1.txt</i></p> <p><i>more diff_1C_1.txt</i></p> <p><i>more diff_ENG_1.txt</i></p> <p>(NB3 : MCX results are not compared due to a problem in reference data)</p>	<p>For each file the content must show that there is no differences using the current precision ( define in file input/precision.tra)</p> <p>The file must contain several records of the following type (x is variable) :</p> <p>-- debut du fichier:output/GRAN_xxx_x/MDRx.tra.diff --</p> <p>-- mot cle recherché xxxxxxxxxxxxxxxx --</p> <p>xxxxxxxxxxxxxxxx NB DIFF <b>NO_DIFF</b></p> <p>-- fin du fichier:output/GRAN_ENG_1/MDR1.tra.diff --</p> <p>....</p>	
--	---	--

Product type	Field / precision	
MDR 1A	GSmcSpect	0,03 K

Product type	Field / precision	
MDR 1B	GS1bSpect	0,03 K

Product type	Field / precision	
MDR 1C	GS1cSpect	0,03 K

Product type	Field / precision	
MDR ENG	MMcxNoiseCalRad	0.01 K Nedt
	MMcxBiasCalRad	0.01 K Nedt
	MMcxFlagNoiseCalRad	égalité
	MMcxFlagBiasCalRad	égalité
	MMcxCoefCalRad	0.001

### 5.2.1.ALGO-OPS-N-15

Test Sheet	ALGO-OPS-N-15	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing in granule mode for the scientific validation (spectrometry post-calibration and spectral calibration for a full orbit)</p> <p>Used data set : J7V3</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>10 Gbytes of data are available.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/ALGO_OPS_N_15</pre> <p>then execute the shell :</p> <pre>./lancetest.sh</pre> <p><b>This shell execute automaticaly the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>		
<p>3. <i>Launch the OPS in standalone mode:</i></p> <pre>.../.../cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetest.sh):</i></p>	The OPS is ready to process input data in nominal mode.	

<p>4. Copy at first, the input auxiliary data, needed to process all granules in the input directory (using symbolic links)</p>		
<p><u>AVHRR product</u> : J0V2 ( see § 3.5.2)</p> <p><u>Other and Stable Parameters , Spectral Database</u> : J7V3 ( see § 3.5.2)</p> <p><u>IASI Context file</u> : J0V2 (see § 3.5.2) this file is copied from the delivery</p> <p><u>Auxiliary data</u> : J0V2 ( see § 3.5.2)</p>		
<p>5. Copy the processing data, needed to process all granule, in the input directory (using symbolic links)</p>	:	
<p><u>working order</u> : IASI_1_wo_001 to IASI_1_wo_041</p> <p><u>IASI L0 data</u> : J7V3 ( see § 3.5.2)</p> <p><u>Product model</u> : J0V2 ( see § 3.5.2) (all granules)</p>		
<p>6. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window for all the granule :</p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p>.....</p> <p>Wait for the processing end.</p>	<p>Verify that each STEP command is acknowledged. (file CMD_xxxx)</p> <p>Verify that with the last granule the processing chain stops (file CMD_xxxx)</p> <p>Verify that the predicted files are produced in the output directory :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul>	
<p>7. Dump the content- of the product files (1A/1B/1C and ENG) and compare with references</p> <p><b>diff.sh ENG x DETAIL</b></p> <p>(NB : x = granule number from 1 to 41)</p> <p><b>diff.sh yy 1 DETAIL</b></p> <p><b>diff.sh yy 23 DETAIL</b></p> <p><b>diff.sh yy 41 DETAIL</b></p> <p>(NB : yy = product type (1A/1B/1C/ENG))</p>	<p>Each product is dumped (1A/1B/1C/ENG) and compared with reference data. Then for a list of fields precised in the files input/keylist_xx.txt (xx=1A, 1B, 1C, ENG) the difference is analysed and summarized in file output/diff_xx.txt (xx=1A, 1B, 1C, ENG)</p> <p>(NB 1: the reference files must have been dumped before using the command</p> <p><b>dumpref.ksh x yy</b>(NB : x = granule number from 1 to 41 and yy = product type (1A/1B/1C/ENG))</p> <p>(NB 2: the precision used to compare the results with the reference is described in several tables here under)</p>	

<p>8. Analyse the differences</p> <p><b>cd output</b></p> <p><b>more diff_ENG_x.txt</b></p> <p>(NB : x = granule number from 1 to 41)</p> <p><b>more diff_yy_1.txt</b></p> <p><b>more diff_yy_23.txt</b></p> <p><b>more diff_yy_41.txt</b></p> <p>(NB: yy = product type (1A/1B/1C/ENG))</p>	<p>For each file the content must show that there is no differences using the current precision ( defined in file input/precision.tra)</p> <p>The file must contain several records of the following type (x is variable) :</p> <pre>-- debut du fichier:output/GRAN_XXX_x/MDRx.tra.diff -- -- mot cle recherché xxxxxxxxxxxxxxxx --       xxxxxxxxxxxxxxxx NB DIFF NO_DIFF -- fin du fichier:output/GRAN_ENG_1/MDR1.tra.diff -- ....</pre> <p><b>NB : at the beginning of the file a summary is done</b></p>	
--	---	--

Product type	Field / precision	
MDR 1A	<b>GrclImage</b>	<b>0.02 K</b>
	<b>GQisFlagQual</b>	<b>égalité</b>
	<b>GQisQualIndex</b>	<b>0,01</b>
	<b>GQisQualIndexIIS</b>	<b>0,01</b>
	<b>GQisQualIndexLoc</b>	<b>0,1</b>
	<b>GQisQualIndexRad</b>	<b>0,01</b>
	<b>GQisQualIndexSpect</b>	<b>0,01</b>
	<b>GSmcSpect</b>	<b>0,03 K</b>

Product type	Field / precision	
MDR 1B	<b>Idem MDR 1A</b>	
MDR 1C	<b>Idem MDR 1A</b>	

Product type	Field / precision	
MDR ENG	BIMSB BT	égalité
	GFtbFilteredBBT	0,001K
	GEPSEndEclipseTime	1 seconde
	GSmeTScan	0,1K
	GSmeFlagDateNok	égalité
	GFtBBBTRes	0,002 K
	GFtBFlagBBTNonQual	égalité
	GSssWnShiftMean	2.35e-3 m-1 (~1.10-8)
	GSssWnShiftMeanQual	1 e-3
	GlaxAxeY	5.10-5 deg
	GlaxAxeZ	5.10-5 deg
	GFaxAxeY	5.10-5 deg
	GFaxAxeZ	5.10-5 deg
	GFaxFlagAxeNonQual	égalité

## 5.2.2.ALGO-OPS-N-16

Test Sheet	ALGO-OPS-N-16	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing in granule mode for the scientific validation (spectrometry post-calibration and spectral calibration for a long term processing with external calibration data)</p> <p>Used data set : J9V3</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>10 Gbytes of data are available.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as <code>iasi_1</code> . You must go in the test directory :</p> <p><code>cd \$WORK_DIR/OPS/data/TV/ALGO_OPS_N_16</code></p> <p>then execute the shell :</p> <p><code>./lancetest.sh</code></p> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <p><code>cd \$WORK_DIR/OPS/tools/OPStools</code></p> <p><i>Clean.sh</i></p>		
<p>3. <i>Launch the OPS in standalone mode:</i></p> <p><code>.../.../cmd/StartStandAlone.sh 1</code></p> <p><code>\$WORK_DIR N</code> (this step is included in the shell <code>lancetest.sh</code>):</p>	The OPS is ready to process input data in nominal mode.	

<p>4. Copy at first, the input auxiliary data, needed to process all granules in the input directory (using symbolic links)</p>		
<p><u>AVHRR product</u> : J0V2 ( see § 3.5.2)</p> <p><u>Other and Stable Parameters , Spectral Database</u> : J9V3 ( see § 3.5.2)</p> <p><u>IASI Context file</u> : J0V2 (see § 3.5.2) this file is copied from the delivery</p> <p><u>Auxiliary data</u> : J0V2 ( see § 3.5.2)</p>		
<p>5. Copy the processing data, needed to process all granule, in the input directory (using symbolic links)</p>	:	
<p><u>working order</u> : IASI_1_wo_001 to IASI_1_wo_041</p> <p><u>IASI L0 data</u> : J9V3 ( see § 3.5.2)</p> <p><u>Product model</u> : J0V2 ( see § 3.5.2) (all granule)</p>		
<p>6. Send then a STEP command to the OPS subsystem :</p> <p>Enter STEP command in the PGF simulator window forall the granule :</p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p>.....</p> <p>Wait for the processing end.</p>	<p>Verify that each STEP command is acknowledged. (file CMD_xxxx)</p> <p>Verify that with the last granule the processing chain stops (file CMD_xxxx)</p> <p>Verify that the predicted files are produced in the output directory :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul>	

<p>7. Dump the content- of the product files (1A/1B/1C and ENG) and compare with references</p> <p><b>diff.sh xx 1 DETAIL</b></p> <p><b>diff.sh xx 2 DETAIL</b></p> <p><b>diff.sh xx 3 DETAIL</b></p> <p><b>diff.sh xx 4 DETAIL</b></p> <p><b>diff.sh xx 5 DETAIL</b></p> <p><b>diff.sh xx 6 DETAIL</b></p> <p><b>diff.sh xx 7 DETAIL</b></p> <p><b>diff.sh xx 8 DETAIL</b></p> <p><b>diff.sh xx 41 DETAIL</b></p> <p>(NB1 : xx = 1A / 1B / 1C / ENG according the product type)</p> <p>(NB2 : 1C granule are not compared for granules 3 5 8)</p> <p>(NB3 : 1A:1B granule are compared for granule 1)</p>	<p>Each product is dumped (1A/1B/1C/ENG) and compared with reference data. Then for a list of fields précised in the files input/keylist_xx.txt (xx=1A, 1B, 1C, ENG) the difference is analysed and summarized in file output/diff_xx.txt (xx=1A, 1B, 1C, ENG)</p> <p>(NB 1: the reference files must have been dumped before using the command</p> <p><b>dumpref.ksh 1</b></p> <p><b>dumpref.ksh 2</b></p> <p><b>dumpref.ksh 3</b></p> <p><b>dumpref.ksh 4</b></p> <p><b>dumpref.ksh 5</b></p> <p><b>dumpref.ksh 6</b></p> <p><b>dumpref.ksh 7</b></p> <p><b>dumpref.ksh 8</b></p> <p><b>dumpref.ksh 41</b></p> <p>(NB 2: the precision used to compare the results with the reference is described in several tables here under)</p>	
<p>8. Analyse the differences</p> <p><b>cd output</b></p> <p><b>cd output</b></p> <p><b>more diff_1A_x.txt</b></p> <p><b>more diff_1B_x.txt</b></p> <p><b>more diff_1C_x.txt</b></p> <p><b>more diff_ENG_x.txt</b></p> <p>(NB1 : xx = 1,2 3,4,5,6,7,8 41 according the granule number)</p> <p>(NB2 : CCS results are not compared)</p>	<p>For each file the content must show that there is no differences using the current precision ( define in file input/precision.tra)</p> <p>The file must contain several records of the following type (x is variable) :</p> <p>-- debut du fichier:output/GRAN_XXX_x/MDRx.tra.diff --</p> <p>-- mot cle recherché xxxxxxxxxxxxxxxx --</p> <p>xxxxxxxxxxxxxxxxxxxxx NB DIFF <b>NO_DIFF</b></p> <p>-- fin du fichier:output/GRAN_ENG_1/MDR1.tra.diff --</p> <p>....</p>	

Product type

Field / precision



Product type	Field / precision	
MDR 1A	DEGRADED_INST_MDR	égalité
	DEGRADED_PROC_MDR	égalité
	GEPSIasiMode	égalité
	GEPSOPSProcessingMode	égalité
	GEPSIdConf	égalité
	GisfLinOrigin	égalité
	GisfColOrigin	égalité
	GisfPds1	2.0e-03
	GisfPds2	2.0e-03
	GisfPds3	2.0e-03
	GisfPds4	2.0e-03
	Girclmage	0.02 K
	GqisFlagQual	égalité
	GqisQualIndex	0,01
	GqisQualIndexIIS	0,01
	GqisQualIndexLoc	0,1
	GqisQualIndexRad	0,01
	GqisQualIndexSpect	0,01
	GSmcSpect	0,03 K

Product type	Field / precision	
MDR 1B	Idem MDR 1A	
MDR 1C	Idem MDR 1A	
	GCcsRadAnalNbClass	0 pour voir la statistique

Product type	Field / precision	
	GccsRadAnalWgt	0 pour voir la statistique
	GccsRadAnalY	0 pour voir la statistique
	GccsRadAnalZ	0 pour voir la statistique
	GCcsRadAnalMean	0 pour voir la statistique
	B0	1.0e-06
	B1_K1	1.0e-01
	B1_K2	1.0e-01
	B1_K3a	1.0e-01
	B1_K3b	0.2
	B1_K5	0.2
	B1_K6	0.2
	B2	1.0e-07
	B3_K1	1.0e-07
	B3_K2	1.0e-07
	B3_K3a	1.0e-07
	B3_K3b	1.0e-07
	B3_K5	1.0e-07
	B3_K6	1.0e-07

Product type	Field / precision	
MDR ENG	BIMSBBT	égalité
	GFtbFilteredBBT	0,001K
	GCcsConfAvhrrChannel	égalité
	GSmeTScan	0,1K
	GSssWnShiftMean	2.35e-3 m-1 (~1.10-8)
	GSssWnShiftMeanQual	1 e-3
	GlaxAxeY	5,0e-5 deg
	GlaxAxeZ	5,0e-5 deg
	GFaxAxeY	5,0e-5 deg
	GFaxAxeZ	5,0e-5 deg
	GFaxFlagAxeNonQual	égalité

Product type	Field / precision	
VIADR_ENG	VERIFICATION_DATA	égalité

### 5.2.3.ALGO-OPS-N-17

Test Sheet	ALGO-OPS-N-17	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing in granule mode for the scientific validation (specific filtering case without history)</p> <p>Used data set : J10V2</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test. 250 Mbytes of data are available.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/ALGO_OPS_N_17</pre> <p>then execute the shell :</p> <pre>./lancetest.sh</pre> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>		
<p>3. <i>Launch the OPS in standalone mode:</i></p> <pre>./.././../cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetest.sh):</i></p>	The OPS is ready to process input data in nominal mode.	
<p>4. <i>Copy at first, the input auxiliary data, needed to process all granules in the input directory (using symbolic links)</i></p>		

AVHRR product : J0V2 ( see § 3.5.2)

Other and Stable Parameters , Spectral Database : J10V2 ( see § 3.5.2)

IASI Context file : J0V2 (see § 3.5.2) this file is copied from the delivery

Auxiliary data : J0V2 ( see § 3.5.2)

5. <i>Copy the processing data, needed to process all granule, in the input directory (using symbolic links)</i>	:	
--	---	--

working order : IASI\_1\_wo\_001

IASI L0 data : J10V2 ( see § 3.5.2)

Product model : J0V2 ( see § 3.5.2) (all granule)

<p>6. Send then a STEP command to the OPS subsystem :</p> <p><i>Enter STEP command in the PGF simulator window for the granule :</i></p> <p><b>STEP 1 input/IASI_1_wo_001</b></p> <p>.....</p> <p><i>Wait for the processing end.</i></p>	<p>Verify that each STEP command is acknowledged. (file CMD_xxxx)</p> <p>Verify that with the last granule the processing chain stops (file CMD_xxxx)</p> <p>Verify that the predicted files are produced in the output directory :</p> <ul style="list-style-type: none"> <li>IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated</li> <li>Report file :</li> </ul>	
<p>7. Dump the content- of the product files (1A/1B/1C and ENG) and compare with references</p> <p><b>diff.sh xx 1 DETAIL</b></p> <p>(NB : xx = 1A / 1B / 1C / ENG according the product type)</p>	<p>Each product is dumped (1A/1B/1C/ENG) and compared with reference data. Then for a list of fields précised in the files input/keylist_xx.txt (xx=1A, 1B, 1C, ENG) the difference is analysed and summarized in file output/diff_xx.txt (xx=1A, 1B, 1C, ENG)</p> <p>(NB 1: the reference files must have been dumped before using the command</p> <p><b>dumpref.ksh 1)</b></p> <p>(NB 2: the precision used to compare the results with the reference is described in several tables here under)</p>	

<p>8. Analyse the differences</p> <p><i>cd output</i></p> <p><i>more diff_1A_1.txt</i></p> <p><i>more diff_1B_1.txt</i></p> <p><i>more diff_1C_1.txt</i></p> <p><i>more diff_ENG_1.txt</i></p>	<p>For each file the content must show that there is no differences using the current precision ( define in file input/precision.tra)</p> <p>The file must contain several records of the following type (x is variable) :</p> <pre>-- debut du fichier:output/GRAN_xxx_x/MDRx.tra.diff -- -- mot cle recherché xxxxxxxxxxxxxxxx --       xxxxxxxxxxxxxxxx NB DIFF <b>NO_DIFF</b> -- fin du fichier:output/GRAN_ENG_1/MDR1.tra.diff -- ....</pre>	
--	---	--

Product type	Field / precision
MDR 1A	

Product type	Field / precision
MDR 1B	

Product type	Field / precision
MDR 1C	

Product type	Field / precision
MDR ENG	

Product type	Field / precision
VIADR_ENG	

## 5.2.4.ALGO-OPS-N-18

Test Sheet	ALGO-OPS-N-18	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing in granule mode for the scientific validation ( processing using three sounder pixels)</p> <p>Used data set : J10V2</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test. 250 Mbytes of data are available.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Connected as iasi_1 . You must go in the test directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/ALGO_OPS_N_18</pre> <p>then execute the shell :</p> <pre>./lancetest.sh</pre> <p><b>This shell execute automaticaly the following steps (in italic style)</b></p>		
<p>2. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>		
<p>3. <i>Launch the OPS in standalone mode:</i></p> <pre>./.././../cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetest.sh):</i></p>	The OPS is ready to process input data in nominal mode.	
<p>4. <i>Copy at first, the input auxiliary data, needed to process all granules in the input directory (using symbolic links)</i></p>		

AVHRR product : J0V2 ( see § 3.5.2)

Other and Stable Parameters , Spectral Database : J11V2 ( see § 3.5.2)

IASI Context file : J0V2 (see § 3.5.2) this file is copied from the delivery

Auxiliary data : J0V2 ( see § 3.5.2)

5. *Copy the processing data, needed to process all granule, in the input directory (using symbolic links)*

:

working order : IASI\_1\_wo\_001

IASI L0 data : J11V2 ( see § 3.5.2)

Product model : J0V2 ( see § 3.5.2) (all granule)

6. Send then a STEP command to the OPS subsystem :

*Enter STEP command in the PGF simulator window for the granule :*

**STEP 1 input/IASI\_1\_wo\_001**

.....

*Wait for the processing end.*

Verify that each STEP command is acknowledged.

(file CMD\_xxxx)

Verify that with the last granule the processing chain stops

(file CMD\_xxxx)

Verify that the predicted files are produced in the output directory :

- IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated
- Report file :

7. Dump the content- of the product files (1A/1B/1C and ENG) and compare with references

**diff.sh xx 1 DETAIL**

(NB : xx = 1A / 1B / 1C / ENG according the product type)

Each product is dumped (1A/1B/1C/ENG) and compared with reference data. Then for a list of fields précised in the files input/keylist\_xx.txt (xx=1A, 1B, 1C, ENG) the difference is analysed and summarized in file output/diff\_xx.txt (xx=1A, 1B, 1C, ENG)

(NB 1: the reference files must have been dumped before using the command

**dumpref.ksh 1)**

(NB 2: the precision used to compare the results with the reference is described in several tables here under)



<p>8. Analyse the differences</p> <p><i>cd output</i></p> <p><i>more diff_1A_1.txt</i></p> <p><i>more diff_1B_1.txt</i></p> <p><i>more diff_1C_1.txt</i></p> <p><i>more diff_ENG_1.txt</i></p>	<p>For each file the content must show that there is no differences using the current precision ( define in file input/precision.tra)</p> <p>The file must contain several records of the following type (x is variable) :</p> <pre>-- debut du fichier:output/GRAN_XXX_x/MDRx.tra.diff -- -- mot cle recherché xxxxxxxxxxxxxxxx --       xxxxxxxxxxxxxxxx NB DIFF <b>NO_DIFF</b> -- fin du fichier:output/GRAN_ENG_1/MDR1.tra.diff -- ....</pre>	
--	---	--

Product type	Field / precision
MDR 1A	

Product type	Field / precision
MDR 1B	

Product type	Field / precision
MDR 1C	

Product type	Field / precision
MDR ENG	

Product type	Field / precision
VIADR_ENG	

## 5.2.5.ALGO-OPS-N-19

Test Sheet	ALGO-OPS-N-19	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the results of the IASI L1 processing in granule mode for the scientific validation</p> <p>Used data set : J0dumpV3</p> <p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>10 Gbytes of data are available.</p>		
Description :	Expected Result :	Tested Requirement:
<p>10. Connected as iasi_1 . You must go in the test directory :</p> <pre>cd \$WORK_DIR/OPS/data/TV/ALGO_OPS_N_19</pre> <p>then execute the shell :</p> <pre>./lancetest.sh</pre> <p><b>This shell execute automatically the following steps (in italic style)</b></p>		
<p>11. <i>Clean the OPS validation environment :</i></p> <pre>cd \$WORK_DIR/OPS/tools/OPStools</pre> <p><i>Clean.sh</i></p>		
<p>12. <i>Launch the OPS in standalone mode:</i></p> <pre>..././cmd/StartStandAlone.sh 1</pre> <p><i>\$WORK_DIR N (this step is included in the shell lancetest.sh):</i></p>	The OPS is ready to process input data in nominal mode.	
<p>13. <i>Copy at first, the input auxiliary data, needed to process the whole granule in the input directory (using symbolic links)</i></p>		

AVHRR product : J0V3 ( see § 3.5.2)

Other and Stable Parameters , Spectral Database : J0V3 ( see § 3.5.2)

IASI Context file : J0V3 (see § 3.5.2) this file is copied from the delivery

Auxiliary data : J0V3 ( see § 3.5.2)

14. *Copy the processing data, needed to process all granule, in the input directory (using symbolic links)*

:

working order : IASI\_1\_wo\_001

IASI L0 data : J0dumpV3 ( see § 3.5.2) ( all granule)

Product model : J0V3 ( see § 3.5.2) (all granule)

15. Send then a STEP command to the OPS subsystem :

*Enter STEP command in the PGF simulator window for the dump :*

**STEP 1 input/IASI\_1\_wo\_001**

*Wait for the processing end.*

Verify that each STEP command is acknowledged.

(file CMD\_xxxx)

Verify that with the last granule the processing chain stops

(file CMD\_xxxx)

Verify that the predicted files are produced in the output directory :

- IASI L1A, 1B 1C ENG , VER named as product model except the processing date which is updated
- Report file :

<p>16. Dump the content- of the product files (1A/1B/1C and ENG) for three lines and compare with references</p> <p>(orbit beginning)</p> <p><i>cd output</i></p> <p><b>TraceProduit “ product file name ” GRAN_xx</b></p> <p>and choose the option to dump MDR for line 2 150 300</p> <p>compare whith reference data</p> <p><i>cd ..</i></p> <p><b>diff.sh xx 1 DUMP</b></p> <p>(NB : xx = 1A / 1B / 1C / ENG according the product type)</p> <p>(NB : y=2, 150 , 300)</p>	<p>Each product is dumped (1A/1B/1C/ENG) and compared with reference data. Then for à list of fields précised in the files input/keylist_xx.txt (xx=1A, 1B, 1C, ENG) the difference is analysed and summarized in file output/diff_xx.txt (xx=1A, 1B, 1C, ENG)</p> <p>(NB 1: the reference files must have been dumped before using the command</p> <p><b>TraceProduit “ product file name ” GRAN_xx )</b></p> <p>(NB 2: the precision used to compare the results with the reference is described in several tables shown in test ALGO-OPS-N-08)</p>	
<p>17. Analyse the differences</p> <p><i>cd output</i></p> <p><b>more diff_1A_1.txt</b></p> <p><b>more diff_1B_1.txt</b></p> <p><b>more diff_1C_1.txt</b></p> <p><b>more diff_ENG_1.txt</b></p>	<p>For each file the content must show that there is no differences using the current precision ( define in file input/precision.tra)</p> <p>The file must contain several records of the following type (x is variable) :</p> <p>-- debut du fichier:output/GRAN_XXX_x/MDRx.tra.diff --</p> <p>-- mot cle recherché xxxxxxxxxxxxxxxx --</p> <p>xxxxxxxxxxxxxxxx NB DIFF <b>NO_DIFF</b></p> <p>-- fin du fichier:output/GRAN_ENG_1/MDR1.tra.diff --</p> <p>....</p>	

## 5.2.6.DBG-OPS-N-01

Test Sheet	DBG-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to test the OPS start-up and execution in investigation mode in the CGS environment.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
<p>1. Replay the GRAN-OPS-N-04 test in investigation mode (first granule)</p> <p><b>export OPS_DEBUG=38_ICC</b></p>	<p>Verify that specific event traces are generated in this mode in tmp/data directory.</p> <p><b>cd output/data</b></p> <p><b>cat DEB*.tra</b></p>	<p>[E_IFS6 pt5]</p> <p>[E_MOD1]</p> <p>[E_DBG1]</p> <p>[PPS-3.2-400]</p> <p>[PPS-3.2-420pt2 et 3]</p>

## 5.2.7.DBG-OPS-N-02

Test Sheet	DBG-OPS-N-02	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to test the OPS processing in investigation mode in the CGS environment.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been installed successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
1. Replay the GRAN-OPS-N-07 test in investigation mode	<p>Verify that specific event traces are generated in this mode.</p> <p>Verify that the investigation log file which are generated during the previous debug test are still available and readable during the current test.</p>	<p><a href="#">[PPS-3.5-060]</a></p> <p>[PPF-0420]</p> <p>[PPS-3.5-070]</p>

**5.2.8.DBG-OPS-D-03**

Test Sheet	DBG-OPS-D-03	FUNCT
<b><u>Purpose :</u></b> The purpose of this test is to test the OPS processing in investigation mode in the CGS environment.		
<b><u>Pre requisites:</u></b> The current OPS version has been install successfully : execution of INST-OPS-N-02 test. If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.		
Description :	Expected Result :	Tested Requirement:
1. Replay the GRAN-OPS-D-13 test in investigation mode	Verify that specific event traces are generated in this mode.	[PPF-0420]

## 5.2.9.STD-OPS-N-01

Test Sheet	STD-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to test the OPS start-up in nominal mode in standalone environment.</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version has been install successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p> <p>Configure the OPS in Standalone (see Installation Manual [DA114]).</p>		
<b>Description :</b>	<b>Expected Result :</b>	<b>Tested Requirement:</b>



<p>1. Launch the OPS subsystem :</p> <p><b>cd \$WORK_DIR/OPS/cmd</b></p> <p><b>StartStandalone.sh 1 \$WORK_DIR N</b></p>	<p>Verify with the</p> <p><b>ps -ef   grep Serveur</b></p> <p><b>ps -ef   grep MainProcess</b></p> <p>commands that the OPS processes are running :</p> <ul style="list-style-type: none"> <li>• MSGS__Serveur ,</li> <li>• MP__MainProcess,</li> <li>• TES__ServeurTemps,</li> <li>• JDBS__Serveur,</li> <li>• SD_FRW__ServeurDonnees,</li> <li>• WOM__ServeurWorkOrder.</li> </ul> <p>Four files, to store CmdTM/StageTM, HKTM status, log events and trace events, are created in the tmp directory as described in chapter 3.2 of Software User Manual [DA112].</p> <p>One UNIX pipe, named cmd_FromPGF is also created in the tmp directory, this pipe is used to send command to the OPS sub system in Standalone mode (see §3.2 of Software User Manual [DA112]).</p> <p>Verify in log and HKTM files that the start of the OPS is reported.</p> <p>Verify that log event are time stamped with 0,1s accuracy.</p>	<p>[SYS-0200]</p> <p>[SYS-0321]</p> <p>[PPS-3.2-080]</p>
<p>2. Stop the OPS : send a STOP commmand :</p> <p><b>echo « STOP 1 »&gt;cmdOPS</b></p>	<p>The OPS is stopped.</p> <p>Verify with the</p> <p><b>ps -ef   grep Serveur</b></p> <p><b>ps -ef   grep MainProcess</b></p> <p>commands that the no OPS process is running.</p> <p>Verify in log files that the stop of the OPS is reported.</p>	<p>[PPF-0350]</p> <p>[PPS-3.2-080]</p>

## 5.2.10.PROC-OPS-N-01

Test Sheet	PROC-OPS-N-01	FUNCT
<p><b><u>Purpose :</u></b></p> <p>The purpose of this test is to validate the availability to process IASI L0 product on a configurable number of thread..</p>		
<p><b><u>Pre requisites:</u></b></p> <p>The current OPS version is install successfully : execution of INST-OPS-N-02 test.</p> <p>If the test is executed in CGS simulated environment, start the CGS simulators : see SIM-OPS-N-01 test.</p>		
Description :	Expected Result :	Tested Requirement:
1. Edit the OPS_SD.cfg configuration file, stored in « config » directory.  Set the “NombreDeThreads” parameter to 2.  Save the modifications.		
2. Launch the OPS.  Execute the 3 first steps of RUN-OPS-N-01 test.	The OPS is ready to process input data in nominal mode.	
3. Replay GRAN_OPS_N_01 test.	Compare thanks to OPS validation tools the IASI L1 generated products with the ones generated by the GRAN_OPS_N_01 test : the products must be similar:	[E_MEL4bis]
4. Edit the OPS_SD.cfg configuration file, stored in « config » directory.  Set the “NombreDeThreads” parameter to 4.  Save the modifications		

5. Clean the OPS validation environment : <b>cd \$WORK_DIR/OPS/tools/OPStools</b> <b>Clean.sh</b>	Verify that : input, to_pgf and tmp directories are purged:  ls \$WORK_ROOT_DIR/input ls \$WORK_ROOT_DIR/to_pgf ls \$WORK_ROOT_DIR/tmp	[SYS-230]
---	--	-----------

## 5.2.11.MULTI-OPS-N-01

The execution of several instances of the same PPF on separated nodes of the SP3 can not be tested in the PGE framework environment. [PPF-0070] [PPS-3.3-010]

## ANNEXE A : FICHE TEST

### Fiche de Test d'Intégration

Fiche de Test	OPS	INT-xxxx-N-01
But du test :		
Conditions initiales requises :		
Description :	Résultat attendu :	
1.		
2. .	.	
3. .		

## Fiche de Test de Validation

Test Sheet	xxxx-OPS-x-xx	xxxxxxxxxx
<b><u>Purpose :</u></b>		
<b><u>Pre requisites:</u></b>		
<b>Description :</b>	<b>Expected Result :</b>	<b>Tested Requirement:</b>

## ANNEXE B : FICHE JOURNAL D'ESSAI

### TEST REPORT

Réf : Ce document a pour but de définir une stratégie d'essais et les procédures de tests	Date : Essai, Validation, IASI, Traitement
---	--

PROJECT : EPS		FACILITY : OPS-IASI		VERSION :	
TEST REPORT IDENTIFIER :			DURATION :		
Author :		Witness :		↑ attached sheet(s)	
Test Result Evaluation :			Witness Signature :		

**TEST REPORT STEP**

Step	SPR	Comments



## ANNEXE C : MATRICE DE COUVERTURE DES INTERFACES

Cette matrice présente la couverture des interfaces par les tests d'intégration.

Interfaces	Tests
f_ParametresAutres	5.1.1-INT-ALGO-N-01 5.1.2-INT-SD-N-01
f_ParametreStables	5.1.1-INT-ALGO-N-01 5.1.2-INT-SD-N-01
f_BanqueSpectrale	5.1.1-INT-ALGO-N-01 5.1.2-INT-SD-N-01
f_DonnéesAuxiliaires	5.1.1-INT-ALGO-N-01 5.1.2-INT-SD-N-01
f_Contexte	5.1.1-INT-ALGO-N-01 5.1.2-INT-SD-N-01 5.1.5-INT-APPL-N-01 5.1.6-INT_APPL_N02
f_ProductsModels	5.1.1-INT-ALGO-N-01 5.1.2-INT-SD-N-01
f_ProduitIASI-In	5.1.1-INT-ALGO-N-01 5.1.2-INT-SD-N-01
f_AlgoTrace	5.1.1-INT-ALGO-N-01 5.1.2-INT-SD-N-01
f_ProduitsIASI-1C	5.1.1-INT-ALGO-N-01
f_DonnéesTechno	5.1.1-INT-ALGO-N-01
Données internes	5.1.2-INT-SD-N-01
f_ProduitsIASI-1A,1B,1C	5.1.2-INT-SD-N-01
f_DonnéesTechnoVerif	5.1.2-INT-SD-N-01 5.1.5-INT-APPL-N-01 5.1.6-INT_APPL_N02
mcs_HKTM	5.1.3-INT-STRUC-N-01 5.1.4-INT-STRUC-D-02 5.1.7-INT-OPS-N-01
m_LogEvent	5.1.3-INT-STRUC-N-01 5.1.4-INT-STRUC-D-02 5.1.5-INT-APPL-N-01 5.1.6-INT_APPL_N02 5.1.7-INT-OPS-N-01
pgf_LogEvent	5.1.3-INT-STRUC-N-01 5.1.4-INT-STRUC-D-02

Interfaces	Tests
	5.1.5-INT-APPL-N-01 5.1.6-INT_APPL_N02 5.1.7-INT-OPS-N-01
m_InitTimer	5.1.3-INT-STRUC-N-01 5.1.5-INT-APPL-N-01
m_EvtTimer	5.1.3-INT-STRUC-N-01 5.1.5-INT-APPL-N-01
pgf_Cmd	5.1.3-INT-STRUC-N-01
pgf_CmdTM	5.1.3-INT-STRUC-N-01 5.1.4-INT-STRUC-D-02 5.1.7-INT-OPS-N-01
m_FinTES	5.1.3-INT-STRUC-N-01 5.1.7-INT-OPS-N-01
m_FinJDBS	5.1.3-INT-STRUC-N-01 5.1.7-INT-OPS-N-01
m_Mort	5.1.4-INT-STRUC-D-02 5.1.5-INT-APPL-N-01 5.1.6-INT_APPL_N02 5.1.7-INT-OPS-N-01
m_CmdState	5.1.5-INT-APPL-N-01
m_FinInit	5.1.5-INT-APPL-N-01 5.1.6-INT_APPL_N02
m_OperationalStatus	5.1.5-INT-APPL-N-01 5.1.6-INT_APPL_N02
f_WorkOrder	5.1.5-INT-APPL-N-01 5.1.6-INT_APPL_N02
m_ProcessingCtrl	5.1.5-INT-APPL-N-01 5.1.6-INT_APPL_N02 5.1.6-INT_APPL_N02
m_Traitement	5.1.5-INT-APPL-N-01 5.1.6-INT_APPL_N02
m_PipelineStatus	5.1.5-INT-APPL-N-01
m_FinTraitement	5.1.5-INT-APPL-N-01 5.1.6-INT_APPL_N02
f_ProduitsIASI1A,1B, 1C	5.1.5-INT-APPL-N-01 5.1.6-INT_APPL_N02
f_Report	5.1.5-INT-APPL-N-01 5.1.6-INT_APPL_N02
m_StageTM	5.1.5-INT-APPL-N-01 5.1.6-INT_APPL_N02
m_FinDump	5.1.5-INT-APPL-N-01 5.1.6-INT_APPL_N02

Interfaces	Tests
m_ArretTraitement	5.1.6-INT_APPL_N02
m_Suspend	5.1.6-INT_APPL_N02
m_Continue	5.1.6-INT_APPL_N02
m_FinSD	5.1.7-INT-OPS-N-01
m_FinWOM	5.1.7-INT-OPS-N-01

## ANNEXE D : MATRICE DE COUVERTURE DES EXIGENCES

Exigences	Paragraphes/Validation
E_CON1	5.2.21-GRAN-OPS-N-06
E_CON2	5.2.21-GRAN-OPS-N-06
E_CON3	5.2.14-CONF-OPS-D-05
E_CON4	5.2.10-GRAN-OPS-N-06
E_CON5	4.3.1.1-Inspection de la Conception
E_CTX0	4.3.1.1-Inspection de la Conception
E_CTX1	4.3.1.1-Inspection de la Conception
E_CTX2	5.2.41-PERF-OPS-N-01
E_CTX3	4.3.1.1-Inspection de la Conception
E_CTX4	5.2.1-GEN-OPS-N-01
E_CTX4bis	4.3.1.1-Inspection de la Conception 4.3.1.2-Inspection du Logiciel
E_CTX5	4.3.1.1-Inspection de la Conception
E_CTX6	4.3.1.1-Inspection de la Conception
E_DBG1	5.2.48-DBG-OPS-N-01
E_DBG2	4.3.1.3-Exigences de Développement
E_DEG0	5.2.25-GRAN-OPS-D-10
E_DEG1	5.2.25-GRAN-OPS-D-10
E_DEG2	5.2.25-GRAN-OPS-D-10 5.2.30-GRAN-OPS-D-15
E_DEG3	5.2.45-ALGO-OPS-N-03
E_DEG5	5.2.15-CONF-OPS-D-06
E_DEG6	5.2.24-GRAN-OPS-N-09
E_DEG7	5.2.25-GRAN-OPS-D-10 5.2.26-GRAN-OPS-D-11 5.2.30-GRAN-OPS-D-15
E_ENC1	4.3.1.1-Inspection de la Conception
E_ENC2	4.3.1.1-Inspection de la Conception
E_ENC4	5.2.43-ALGO-OPS-N-01
E_FIC1	5.2.10-CONF-OPS-N-01
E_FIC2	5.2.10-CONF-OPS-N-01
E_FIC3	5.2.10-CONF-OPS-N-01
E_FIC4	5.2.10-CONF-OPS-N-01
E_FIC5	5.2.22-GRAN-OPS-N-07
E_FIC6	5.2.22-GRAN-OPS-N-07
E_FLA1	5.2.25-GRAN-OPS-D-10
E_IFS1	4.3.1.1-Inspection de la Conception
E_IFS10	5.2.19-GRAN-OPS-N-04
E_IFS2ter	5.2.19-GRAN-OPS-N-04
E_IFS3	5.2.19-GRAN-OPS-N-04
E_IFS4	4.3.1.1-Inspection de la Conception
E_IFS4bis	4.3.1.1-Inspection de la Conception
E_IFS5	4.3.1.1-Inspection de la Conception
E_IFS6	4.3.1.1-Inspection de la Conception
E_IFS6 pt2	5.2.16-GRAN-OPS-N-01
E_IFS6 pt3	5.2.16-GRAN-OPS-N-01
E_IFS6 pt5	5.2.16-GRAN-OPS-N-01

Exigences	Paragraphes/Validation
	5.2.48-DBG-OPS-N-01
E_IFS7	4.3.1.1-Inspection de la Conception
E_IFS8	5.2.31-GRAI-OPS-N-01 5.2.32-GRAI-OPS-N-02
E_IFS9	5.2.19-GRAN-OPS-N-04 5.2.33-DUMP-OPS-N-01 5.2.42-PERF-OPS-N-02 5.2.42-PERF-OPS-N-02
E_INI1	4.3.1.1-Inspection de la Conception
E_INI2	5.2.10-CONF-OPS-N-01
E_INI3	5.2.10-CONF-OPS-N-01
E_INI4	5.2.10-CONF-OPS-N-01
E_INI5	5.2.1-GEN-OPS-N-01
E_INI6	4.3.1.1-Inspection de la Conception
E_INT1	5.2.28-GRAN-OPS-D-13
E_INT10	5.2.45-ALGO-OPS-N-03
E_INT11	5.2.37-LOG-OPS-N-01
E_INT12	4.3.1.3-Exigences de Développement
E_INT13	5.2.37-LOG-OPS-N-01
E_INT2	5.2.44-ALGO-OPS-N-02
E_INT3	4.3.1.1-Inspection de la Conception
E_INT4	4.3.1.1-Inspection de la Conception
E_INT5	4.3.1.1-Inspection de la Conception
E_INT6	4.3.1.1-Inspection de la Conception
E_INT7	4.3.1.1-Inspection de la Conception
E_INT8	5.2.34-OVER-OPS-N-01
E_INT8bis	5.2.35-OVER-OPS-D-02
E_INT9	5.2.19-GRAN-OPS-N-04
E_LIB1	4.3.1.1-Inspection de la Conception
E_LIB2	4.3.1.1-Inspection de la Conception
E_LIB3	4.3.1.1-Inspection de la Conception
E_LIB3bis	4.3.1.1-Inspection de la Conception
E_LIB4	4.3.1.1-Inspection de la Conception
E_MEL1	4.3.1.1-Inspection de la Conception 4.3.1.3-Exigences de Développement
E_MEL1bis	4.3.1.1-Inspection de la Conception 4.3.1.3-Exigences de Développement
E_MEL2	4.3.1.2-Inspection du Logiciel
E_MEL3	4.3.1.1-Inspection de la Conception
E_MEL4	4.3.1.1-Inspection de la Conception
E_MEL4bis	5.2.52-PROC-OPS-N-01
E_MEL4quar	4.3.1.1-Inspection de la Conception
E_MEL4ter	4.3.1.1-Inspection de la Conception
E_MEL7	4.3.1.2-Inspection du Logiciel
E_MEL8	4.3.1.1-Inspection de la Conception
E_MOD1	5.2.23-GRAN-OPS-N-08 5.2.24-GRAN-OPS-N-09 5.2.31-GRAI-OPS-N-01 5.2.33-DUMP-OPS-N-01 5.2.48-DBG-OPS-N-01
E_MOD2	5.2.45-ALGO-OPS-N-03
E_MOD3	5.2.33-DUMP-OPS-N-01
E OPC0	4.3.1.2-Inspection du Logiciel

Exigences	Paragraphes/Validation
E OPC0bis	4.3.1.2-Inspection du Logiciel
E OPC1	4.3.1.2-Inspection du Logiciel
E OPC2	4.3.1.1-Inspection de la Conception
E OPC3	4.3.1.2-Inspection du Logiciel
E OPC4	4.3.1.2-Inspection du Logiciel
E OPC5	4.3.1.2-Inspection du Logiciel
E OPC6	4.3.1.2-Inspection du Logiciel
E OPC7	4.3.1.2-Inspection du Logiciel
E OPC8	4.3.1.2-Inspection du Logiciel
E OPC9	4.3.1.2-Inspection du Logiciel
E OPI1	4.3.1.1-Inspection de la Conception
E OPI2	4.3.1.1-Inspection de la Conception
E OPI3	4.3.2.3-Non Conformes
E OPR1	4.3.1.1-Inspection de la Conception
E RET1	5.2.31-GRAI-OPS-N-01
E SEQ1	5.2.45-ALGO-OPS-N-03
E SEQ2	5.2.45-ALGO-OPS-N-03
E SEQ4	5.2.45-ALGO-OPS-N-03
E SEQ5	5.2.45-ALGO-OPS-N-03
E STR1	4.3.1.2-Inspection du Logiciel
E TST1	4.3.1.3-Exigences de Développement
E TST2	4.3.1.3-Exigences de Développement
E TST3	4.3.1.3-Exigences de Développement
E TST4	4.3.1.3-Exigences de Développement
E TST5	4.3.1.3-Exigences de Développement
E TST6	4.3.1.3-Exigences de Développement
E TST7	4.3.1.3-Exigences de Développement
E TST8	4.3.1.3-Exigences de Développement
IF-ALGO-4.1-100	5.2.17-GRAN-OPS-D-02
IF-ALGO-4.1-110	4.3.1.1-Inspection de la Conception
IF-ALGO-4.1-120	4.3.2.1-Exigences Non Applicables
IF-ALGO-4.1-130	4.3.2.2-Non Implémentées par le PGE
IF-ALGO-4.1-140	4.3.1.1-Inspection de la Conception
IF-ALGO-4.2-100	5.2.7-RUN-OPS-N-01 5.2.8-RUN-OPS-D-02 5.2.39-CMD-OPS-N-01
IF-ALGO-4.2-110	4.3.2.1-Exigences Non Applicables
IF-ALGO-4.2-130	4.3.2.1-Exigences Non Applicables
IF-ALGO-4.2-140	5.2.7-RUN-OPS-N-01 5.2.39-CMD-OPS-N-01
IF-ALGO-4.2-150	4.3.2.1-Exigences Non Applicables
IF-ALGO-4.2-160	4.3.2.1-Exigences Non Applicables
IF-ALGO-4.2-170	4.3.2.1-Exigences Non Applicables
IF-ALGO-4.3-100	4.3.1.1-Inspection de la Conception
IF-ALGO-4.3-110	4.3.2.1-Exigences Non Applicables
IF-ALGO-4.4-100	4.3.2.2-Non Implémentées par le PGE
IF-ALGO-4.4-110	4.3.2.2-Non Implémentées par le PGE
IF-ALGO-4.4-120	4.3.2.2-Non Implémentées par le PGE
IF-ALGO-4.4-130	4.3.1.1-Inspection de la Conception
IF-ALGO-4.4-131	5.2.43-ALGO-OPS-N-01
IF-ALGO-4.4-140	5.2.43-ALGO-OPS-N-01
IF-ALGO-4.4-160	5.2.43-ALGO-OPS-N-01

Exigences	Paragraphes/Validation
IF-ALGO-4.4-170	5.2.43-ALGO-OPS-N-01
IF-ALGO-4.4-180	5.2.43-ALGO-OPS-N-01
IF-ALGO-4.4-181	4.3.2.1-Exigences Non Applicables 4.3.2.2-Non Implémentées par le PGE
IF-ALGO-4.4-182	4.3.2.2-Non Implémentées par le PGE
IF-ALGO-4.4-183	5.2.43-ALGO-OPS-N-01
IF-ALGO-4.4-190	4.3.2.1-Exigences Non Applicables 4.3.2.2-Non Implémentées par le PGE
IF-ALGO-4.4-200	4.3.2.1-Exigences Non Applicables 4.3.2.2-Non Implémentées par le PGE
PPF-0010	5.2.19-GRAN-OPS-N-04
PPF-0030	5.2.19-GRAN-OPS-N-04
PPF-0040	5.2.4-INST-OPS-N-03
PPF-0050	4.3.1.1-Inspection de la Conception
PPF-0060	5.2.19-GRAN-OPS-N-04 5.2.31-GRAI-OPS-N-01 5.2.32-GRAI-OPS-N-02
PPF-0070	5.2.53-MULTI-OPS-N-01
PPF-0080	5.2.19-GRAN-OPS-N-04
PPF-0090	4.3.2.1-Exigences Non Applicables
PPF-0100	4.3.2.1-Exigences Non Applicables
PPF-0110	4.3.1.3-Exigences de Développement
PPF-0120	5.2.26-GRAN-OPS-D-11
PPF-0140	5.2.16-GRAN-OPS-N-01
PPF-0160	5.2.16-GRAN-OPS-N-01
PPF-0200	5.2.41-PERF-OPS-N-01
PPF-0210	5.2.41-PERF-OPS-N-01
PPF-0218	4.3.2.1-Exigences Non Applicables
PPF-0220	4.3.1.1-Inspection de la Conception
PPF-0230	4.3.1.1-Inspection de la Conception
PPF-0250	4.3.2.1-Exigences Non Applicables
PPF-0271	5.2.23-GRAN-OPS-N-08
PPF-0300	4.3.2.1-Exigences Non Applicables
PPF-0310	5.2.16-GRAN-OPS-N-01
PPF-0340	5.2.43-ALGO-OPS-N-01
PPF-0350	5.2.7-RUN-OPS-N-01 5.2.39-CMD-OPS-N-01
PPF-0370	4.3.1.2-Inspection du Logiciel
PPF-0390	4.3.1.1-Inspection de la Conception
PPF-0390pt1	5.2.7-RUN-OPS-N-01 5.2.40-HKTM-OPS-N-01
PPF-0400 pt 2,3,4	5.2.37-LOG-OPS-N-01
PPF-0400 pt 4	5.2.37-LOG-OPS-N-01
PPF-0400pt 1	5.2.35-OVER-OPS-D-02
PPF-0410	4.3.1.1-Inspection de la Conception
PPF-0420	5.2.28-GRAN-OPS-D-13 5.2.49-DBG-OPS-N-02
PPF-0430	5.2.8-RUN-OPS-D-02
PPF-0440	5.2.16-GRAN-OPS-N-01
PPF-0460	5.2.28-GRAN-OPS-D-13
PPF-0470	5.2.28-GRAN-OPS-D-13
PPF-0480	5.2.28-GRAN-OPS-D-13
PPF-0490	5.2.46-ALGO-OPS-N-04

Exigences	Paragraphes/Validation
PPF-0500	5.2.19-GRAN-OPS-N-04
PPF-0510	5.2.19-GRAN-OPS-N-04
PPF-0520	4.3.2.1-Exigences Non Applicables
PPF-0530	5.2.19-GRAN-OPS-N-04
PPF-0530 pt2	5.2.41-PERF-OPS-N-2
PPF-0550	5.2.16-GRAN-OPS-N-01
PPF-0554	5.2.4-INST-OPS-N-03
PPF-0580	5.2.38-ERR-OPS-N-01
PPF-0610pt1	5.2.35-OVER-OPS-D-02
PPF-0610pt2	4.3.1.1-Inspection de la Conception
PPF-0610pt3	5.2.11-CONF-OPS-D-02 5.2.12-CONF-OPS-D-03 5.2.13-CONF-OPS-D-04
PPF-0630pt2	5.2.23-GRAN-OPS-N-08 5.2.31-GRAI-OPS-N-01
PPF-0630pt3	5.2.21-GRAN-OPS-N-06
PPF-0640	5.2.34-OVER-OPS-N-01
PPF-0641	5.2.25-GRAN-OPS-D-10
PPF-0670	4.3.2.1-Exigences Non Applicables
PPF-0681	5.2.39-CMD-OPS-N-01
PPF-0690	5.2.7-RUN-OPS-N-01
PPF-0691	5.2.19-GRAN-OPS-N-04
PPF-0692	5.2.16-GRAN-OPS-N-01
PPF-0693	5.2.39-CMD-OPS-N-01
PPF-0700	5.2.14-CONF-OPS-D-05
PPF-0701	4.3.2.1-Exigences Non Applicables
PPF-0710	4.3.1.1-Inspection de la Conception
PPF-0750	4.3.1.1-Inspection de la Conception
PPF-0760	4.3.1.3-Exigences de Développement
PPF-0770	5.2.7-RUN-OPS-N-01
PPF-0780	4.3.1.1-Inspection de la Conception 4.3.2.2-Non Implémentées par le PGE
PPF-0790	4.3.1.1-Inspection de la Conception
PPF-0800	4.3.1.1-Inspection de la Conception
PPF-0810	4.3.1.1-Inspection de la Conception
PPF-0811	4.3.2.1-Exigences Non Applicables
PPF-0820	5.2.19-GRAN-OPS-N-04 5.2.21-GRAN-OPS-N-06 5.2.33-DUMP-OPS-N-01
PPF-0830	4.3.2.4-Non Testables
PPF-0840	5.2.21-GRAN-OPS-N-06
PPF-0850pt1	5.2.21-GRAN-OPS-N-06
PPF-0850pt2	5.2.27-GRAN-OPS-D-12
PPF-0860	4.3.1.1-Inspection de la Conception
PPF-0870	5.2.30-GRAN-OPS-D-15
PPF-0880	5.2.21-GRAN-OPS-N-06
PPF-0890	4.3.1.1-Inspection de la Conception
PPF-0900	4.3.1.1-Inspection de la Conception
PPF-0910	5.2.37-LOG-OPS-N-01
PPF-0920	4.3.1.1-Inspection de la Conception
PPF-0940	4.3.1.1-Inspection de la Conception
PPF-0950	4.3.1.1-Inspection de la Conception
PPF-0960	5.2.4-INST-OPS-N-03



Exigences	Paragraphes/Validation
PPF-0970	4.3.2.1-Exigences Non Applicables
PPS-3.1-030	4.3.1.1-Inspection de la Conception
PPS-3.1-050	4.3.1.1-Inspection de la Conception
PPS-3.1-060	4.3.1.1-Inspection de la Conception
PPS-3.1-070	4.3.1.1-Inspection de la Conception
PPS-3.2-010	5.2.39-CMD-OPS-N-01
PPS-3.2-020	4.3.2.2-Non Implémentées par le PGE
PPS-3.2-030	5.2.7-RUN-OPS-N-01 5.2.39-CMD-OPS-N-01
PPS-3.2-040	5.2.16-GRAN-OPS-N-01
PPS-3.2-050	4.3.1.1-Inspection de la Conception
PPS-3.2-060	5.2.16-GRAN-OPS-N-01
PPS-3.2-070	4.3.1.1-Inspection de la Conception
PPS-3.2-080	5.2.51-STD-OPS-N-01
PPS-3.2-090	5.2.25-GRAN-OPS-D-10 5.2.26-GRAN-OPS-D-11 5.2.30-GRAN-OPS-D-15
PPS-3.2-100	5.2.26-GRAN-OPS-D-11 5.2.37-LOG-OPS-N-01
PPS-3.2-110 pt 5 et 6	5.2.19-GRAN-OPS-N-04
PPS-3.2-110 pt1,2,3,4,7	5.2.16-GRAN-OPS-N-01 5.2.16-GRAN-OPS-N-01 5.2.16-GRAN-OPS-N-01
PPS-3.2-130	4.3.2.1-Exigences Non Applicables
PPS-3.2-140	4.3.2.1-Exigences Non Applicables
PPS-3.2-150	4.3.2.1-Exigences Non Applicables
PPS-3.2-160	5.2.23-GRAN-OPS-N-08 5.2.24-GRAN-OPS-N-09
PPS-3.2-170	5.2.24-GRAN-OPS-N-09
PPS-3.2-310	5.2.12-CONF-OPS-D-03
PPS-3.2-320	5.2.21-GRAN-OPS-N-06
PPS-3.2-330	4.3.2.1-Exigences Non Applicables
PPS-3.2-350	5.2.31-GRAI-OPS-N-01 5.2.32-GRAI-OPS-N-02
PPS-3.2-370	4.3.2.1-Exigences Non Applicables
PPS-3.2-400	5.2.16-GRAN-OPS-N-01 5.2.48-DBG-OPS-N-01
PPS-3.2-420pt2 et 3	5.2.31-GRAI-OPS-N-01 5.2.48-DBG-OPS-N-01
PPS-3.2-440	5.2.21-GRAN-OPS-N-06
PPS-3.2-490	5.2.28-GRAN-OPS-D-13 5.2.29-GRAN-OPS-D-14
PPS-3.2-510	4.3.2.1-Exigences Non Applicables
PPS-3.3-010	5.2.53-MULTI-OPS-N-01
PPS-3.3-020	4.3.2.2-Non Implémentées par le PGE
PPS-3.3-030	4.3.1.1-Inspection de la Conception
PPS-3.3-040	4.3.1.1-Inspection de la Conception
PPS-3.3-050	4.3.1.1-Inspection de la Conception
PPS-3.3-060	4.3.1.1-Inspection de la Conception
PPS-3.3-070	4.3.2.1-Exigences Non Applicables
PPS-3.3-080	4.3.1.1-Inspection de la Conception
PPS-3.3-090	4.3.1.2-Inspection du Logiciel
PPS-3.3-110pt1	4.3.1.1-Inspection de la Conception

Exigences	Paragraphes/Validation
PPS-3.3-110pt2	5.2.15-CONF-OPS-D-06
PPS-3.3-130	4.3.1.1-Inspection de la Conception
PPS-3.4-020	4.3.2.1-Exigences Non Applicables
PPS-3.4-030	5.2.16-GRAN-OPS-N-01
PPS-3.4-040	5.2.10-CONF-OPS-N-01
PPS-3.4-060	4.3.2.1-Exigences Non Applicables
PPS-3.4-070	4.3.2.1-Exigences Non Applicables
PPS-3.4-080	4.3.2.1-Exigences Non Applicables
PPS-3.4-090	4.3.2.1-Exigences Non Applicables
PPS-3.4-100	4.3.2.1-Exigences Non Applicables
PPS-3.5-020	4.3.2.4-Non Testables
PPS-3.5-030	4.3.2.4-Non Testables
PPS-3.5-035pt1	5.2.42-PERF-OPS-N-02
PPS-3.5-035pt2	4.3.2.4-Non Testables
PPS-3.5-040	5.2.4-INST-OPS-N-03
PPS-3.5-050	5.2.28-GRAN-OPS-D-13
PPS-3.5-060	5.2.49-DBG-OPS-N-02
PPS-3.5-070	5.2.49-DBG-OPS-N-02
PPS-3.5-090	4.3.1.1-Inspection de la Conception
PPS-3.5-100	4.3.1.1-Inspection de la Conception
PPS-3.5-110	4.3.1.1-Inspection de la Conception
PPS-3.5-120	4.3.2.1-Exigences Non Applicables
PPS-3.6-010	4.3.2.1-Exigences Non Applicables
PPS-3.6-030	5.2.20-GRAN-OPS-N-05 5.2.21-GRAN-OPS-N-06 5.2.23-GRAN-OPS-N-08
PPS-3.6-040	5.2.42-PERF-OPS-N-02
PPS-3.6-050	5.2.20-GRAN-OPS-N-05 5.2.33-DUMP-OPS-N-01
PPS-3.6-060	4.3.1.4-Analyse des Résultats des Tests de validation
PPS-3.6-070	5.2.4-INST-OPS-N-03
PPS-3.7-010	4.3.1.4-Analyse des Résultats des Tests de validation
PPS-3.7-020	4.3.1.3-Exigences de Développement
PPS-3.7-020 pt2	4.3.2.1-Exigences Non Applicables
PPS-3.7-030	4.3.2.1-Exigences Non Applicables
PPS-3.7-050	4.3.1.3-Exigences de Développement
PPS-3.7-080	4.3.1.3-Exigences de Développement
PPS-3.7-090	5.2.2-INST-OPS-N-01
SYS-0010	4.3.2.4-Non Testables
SYS-0020	4.3.2.4-Non Testables
SYS-0030	4.3.1.1-Inspection de la Conception
SYS-0040	4.3.2.1-Exigences Non Applicables
SYS-0050	4.3.2.1-Exigences Non Applicables
SYS-0060	5.2.16-GRAN-OPS-N-01
SYS-0070	4.3.1.1-Inspection de la Conception
SYS-0080 bullet3	5.2.2-INST-OPS-N-01
SYS-0080 pt1	5.2.16-GRAN-OPS-N-01
SYS-0090	4.3.2.1-Exigences Non Applicables
SYS-0100	4.3.2.1-Exigences Non Applicables
SYS-0110	5.2.2-INST-OPS-N-01
SYS-0120	5.2.2-INST-OPS-N-01
SYS-0130	5.2.2-INST-OPS-N-01
SYS-0140	5.2.2-INST-OPS-N-01

Exigences	Paragraphes/Validation
SYS-0150	5.2.1-GEN-OPS-N-01
SYS-0160	4.3.2.1-Exigences Non Applicables
SYS-0161	5.2.4-INST-OPS-N-03
SYS-0162	4.3.2.1-Exigences Non Applicables
SYS-0170	4.3.2.1-Exigences Non Applicables
SYS-0171	4.3.2.1-Exigences Non Applicables
SYS-0172	4.3.2.1-Exigences Non Applicables
SYS-0173	4.3.2.1-Exigences Non Applicables
SYS-0173bis	4.3.2.1-Exigences Non Applicables
SYS-0174	4.3.2.1-Exigences Non Applicables
SYS-0180	4.3.2.1-Exigences Non Applicables
SYS-0190	4.3.2.1-Exigences Non Applicables
SYS-0200	5.2.51-STD-OPS-N-01
SYS-0210	4.3.2.1-Exigences Non Applicables
SYS-0220	4.3.1.1-Inspection de la Conception
SYS-0230	4.3.1.1-Inspection de la Conception 5.2.53-MULTI-OPS-N-01 (AC)
SYS-0240	4.3.2.1-Exigences Non Applicables
SYS-0250	4.3.1.1-Inspection de la Conception
SYS-0260	4.3.1.1-Inspection de la Conception
SYS-0270	4.3.1.1-Inspection de la Conception
SYS-0280	4.3.1.1-Inspection de la Conception
SYS-0290	5.2.7-RUN-OPS-N-01
SYS-0300	4.3.1.1-Inspection de la Conception
SYS-0310	4.3.1.1-Inspection de la Conception
SYS-0320	4.3.1.1-Inspection de la Conception
SYS-0321	5.2.51-STD-OPS-N-01
SYS-0330	5.2.16-GRAN-OPS-N-01
SYS-0330 category P et O	5.2.7-RUN-OPS-N-01 5.2.16-GRAN-OPS-N-01
SYS-0340	5.2.16-GRAN-OPS-N-01
SYS-0350	5.2.16-GRAN-OPS-N-01
SYS-0360	4.3.2.1-Exigences Non Applicables
SYS-0370	5.2.16-GRAN-OPS-N-01
SYS-0380	5.2.7-RUN-OPS-N-01 5.2.8-RUN-OPS-D-02
SYS-0390	5.2.7-RUN-OPS-N-01 5.2.39-CMD-OPS-N-01
SYS-0400	4.3.2.1-Exigences Non Applicables
SYS-0410	4.3.2.1-Exigences Non Applicables
SYS-0420	4.3.2.1-Exigences Non Applicables
SYS-0430	4.3.2.1-Exigences Non Applicables
SYS-0440	5.2.9-RUN-OPS-D-03 5.2.16-GRAN-OPS-N-01 5.2.39-CMD-OPS-N-01
SYS-0450	4.3.2.1-Exigences Non Applicables
SYS-0460	4.3.2.1-Exigences Non Applicables
SYS-0470	4.3.2.1-Exigences Non Applicables
SYS-0480	4.3.1.1-Inspection de la Conception 5.2.37-LOG-OPS-N-01 5.2.37-LOG-OPS-N-01
SYS-0490	5.2.7-RUN-OPS-N-01 5.2.16-GRAN-OPS-N-01 5.2.39-CMD-OPS-N-01

Exigences	Paragraphes/Validation
SYS-0500	4.3.1.1-Inspection de la Conception
SYS-0510	4.3.2.1-Exigences Non Applicables
SYS-0520	4.3.2.1-Exigences Non Applicables
SYS-0521	4.3.2.1-Exigences Non Applicables
SYS-0522	4.3.2.1-Exigences Non Applicables
SYS-0523	4.3.2.1-Exigences Non Applicables
SYS-0530	4.3.2.1-Exigences Non Applicables
SYS-0531	4.3.2.1-Exigences Non Applicables
SYS-0540	4.3.2.1-Exigences Non Applicables
SYS-0550	4.3.2.1-Exigences Non Applicables
SYS-0560	4.3.2.1-Exigences Non Applicables
SYS-0570	4.3.2.1-Exigences Non Applicables
SYS-0571	4.3.2.1-Exigences Non Applicables
SYS-0580	4.3.2.1-Exigences Non Applicables
SYS-0590	4.3.2.1-Exigences Non Applicables
SYS-0600	4.3.2.1-Exigences Non Applicables
SYS-0610	4.3.2.1-Exigences Non Applicables
SYS-0620	4.3.2.1-Exigences Non Applicables
SYS-0630	4.3.2.1-Exigences Non Applicables
SYS-0640	4.3.2.1-Exigences Non Applicables
SYS-0641	4.3.2.1-Exigences Non Applicables
SYS-0650	4.3.2.1-Exigences Non Applicables
SYS-0660	4.3.2.1-Exigences Non Applicables
SYS-0670	4.3.2.1-Exigences Non Applicables
SYS-0671	4.3.2.1-Exigences Non Applicables
SYS-0680	4.3.1.1-Inspection de la Conception
SYS-0690	4.3.1.1-Inspection de la Conception
SYS-0700	4.3.1.1-Inspection de la Conception
SYS-0710	4.3.2.1-Exigences Non Applicables
SYS-0720	4.3.1.1-Inspection de la Conception
SYS-0730	5.2.2-INST-OPS-N-01
SYS-0740	4.3.1.1-Inspection de la Conception
SYS-0750	4.3.2.1-Exigences Non Applicables
SYS-0760	4.3.1.1-Inspection de la Conception
SYS-0770	4.3.2.1-Exigences Non Applicables
SYS-0780	4.3.1.1-Inspection de la Conception
SYS-0790	4.3.1.1-Inspection de la Conception
SYS-0800	4.3.1.1-Inspection de la Conception
SYS-0810	4.3.1.1-Inspection de la Conception
SYS-0820	4.3.1.1-Inspection de la Conception
SYS-0830	4.3.1.1-Inspection de la Conception
SYS-0840	4.3.1.1-Inspection de la Conception
SYS-0850	4.3.1.1-Inspection de la Conception
SYS-0860	4.3.1.1-Inspection de la Conception
SYS-0900	4.3.2.1-Exigences Non Applicables
SYS-0910	4.3.2.1-Exigences Non Applicables
SYS-0920	4.3.2.1-Exigences Non Applicables
SYS-0930	4.3.2.1-Exigences Non Applicables
SYS-0940	4.3.2.1-Exigences Non Applicables
SYS-0950	4.3.2.1-Exigences Non Applicables
SYS-0960	4.3.2.1-Exigences Non Applicables
SYS-0970	4.3.2.1-Exigences Non Applicables

Exigences	Paragraphes/Validation
SYS-0980	4.3.2.1-Exigences Non Applicables
SYS-0990	4.3.2.1-Exigences Non Applicables
SYS-1000 => 1730	4.3.2.1-Exigences Non Applicables