



Guide de déploiement et d'utilisation du logiciel de transfert de données et métadonnées validées

version 1.0

S

Historique des modifications

| Date | Version | Description | Auteur |
|--------------|---------|--|-------------------------|
| 19 Juin 2012 | 0.1 | Écriture initiale (maquette non fonctionnelle du logiciel) | P.Völcke |
| 3 Oct 2013 | 0.2 | Refonte du document (prototype fonctionnel du logiciel) | P.Völcke C.Pequegnat |
| 20 Déc 2013 | 0.3 | Modifications suites à suggestions des centres de collecte (ajouts d'informations liées à l'intégration des données sismologiques validées + précisions sur le <i>workflow</i>) | P.Völcke C.Pequegnat |
| 18 Mars 2014 | 1.0 | Mise en production du système | P.Völcke C.Pequegnat |
| 24 Mars 2014 | 1.0.1 | Modification de description du processus T3 (intégration de données) | C.Pequegnat |
| Août 2014 | 1.0.2 | Précision sur la collecte du numéro de transaction par l'appelant (4.5). | P.Völcke |

Sommaire

| | |
|---|---|
| Historique des modifications..... | 2 |
| Sommaire..... | 2 |
| 1 Introduction..... | 3 |
| 2 Documents de référence..... | 3 |
| 3 Guide d'installation, mise à jour, et configuration..... | 3 |
| 4 Guide d'exploitation..... | 3 |
| 4.1 Rappels..... | 3 |
| 4.2 Fonctionnement général du logiciel, aide en ligne..... | 4 |
| 4.3 Transmettre des données sismologiques validées..... | 4 |
| 4.4 Transmettre des métadonnées sismologiques validées..... | 4 |
| 4.5 Obtenir un identifiant de transaction, lire les sorties standards et le journal de transaction..... | 5 |
| 4.6 Obtenir et analyser un état de transaction..... | 5 |
| Annexes..... | 7 |
| Format de l'identifiant de transaction..... | 7 |
| Etat d'intégration de données validées : structure du fichier XML..... | 7 |
| Etat d'intégration de métadonnées validées : structure du fichier XML..... | 9 |
| Format du journal de transaction..... | 9 |

1 Introduction

Ce document s'adresse aux opérateurs des noeuds de collecte et validation (« *noeuds A* ») du système d'information RESIF. Il décrit l'installation, la configuration, et l'utilisation du logiciel de transfert de données vers le noeud de distribution (« *noeud B* ») du système d'information RESIF.

Le logiciel *ResifDataTransfer* est mis à disposition des noeuds de collecte et validation RESIF afin de transmettre leurs données validées et de demander leur intégration dans le système d'information RESIF. Ce processus comprend plusieurs étapes:

- le transfert des données d'un *noeud A* vers le *noeud B*
- la vérification des données reçues par le *noeud B*
- l'intégration de données dans les différentes bases de données du *noeud B*
- la mise à disposition auprès du *noeud A* d'un journal d'intégration des données

2 Documents de référence

« DCTP Transmission de données validées des noeuds A vers le noeud B »

https://resif.dt.insu.cnrs.fr/fileviewer.php?file_id=259

« DCTP Types de données validées et conditions de leur intégration dans le noeud B »

https://resif.dt.insu.cnrs.fr/fileviewer.php?file_id=296

Note : la langue des documents de référence est le Français. Lorsque jugé utile, la langue utilisée dans les aides en ligne des logiciels est l'Anglais, afin de permettre l'utilisation quotidienne par des opérateurs non francophones.

3 Guide d'installation, mise à jour, et configuration

Se reporter aux instructions contenues dans les fichiers *INSTALL* et *UPGRADE*, ainsi que dans le modèle de fichier de configuration (*ResifDataTransfer.conf.dist*).

4 Guide d'exploitation

4.1 Rappels

Cette section reprend les principaux éléments des documents de références (voir page 3).

Le principe général de la transmission de données est une transmission par lots (ou **transaction**). Une transaction est constituée :

- d'un ensemble de fichiers transmis par un noeud de collecte et validation (un seul dans le cas d'un transfert de métadonnées sismologiques validées)
- d'un type de données associé à cet ensemble de fichiers (par exemple : *données sismologiques validées au format miniseed*)
- d'un état détaillé de la transaction. Cet état détaille le statut d'intégration des fichiers transmis au centre de distribution, selon les critères qualitatifs définis dans les documents de référence. Cet état prend la forme d'un fichier au format *XML*. La structure du fichier associé à une transaction est décrite en Annexes.

4.2 Fonctionnement général du logiciel, aide en ligne

Le logiciel est constitué :

- d'un fichier exécutable *ResifDataTransfer.py* (non modifiable)
- d'un fichier de configuration textuel nommé librement, ex : *ResifDataTransfer.conf* , modifiable par l'opérateur

L'exécution du logiciel s'effectue en ligne de commande. Il est ainsi possible d'automatiser une partie des tâches de transfert (ex : *cron*).

Une aide en ligne est disponible. Celle-ci contient les informations les plus à jour :

```
$ ResifDataTransfer.py --help
```

Les sections suivantes présentent des cas d'utilisation courant du logiciel. Dans la plupart des exemples présentés, les options peuvent être exprimées sous forme longue (ex : *--option*) ou courte (ex : *-o*).

4.3 Transmettre des données sismologiques validées

Désigner au logiciel de transfert un répertoire contenant les fichiers à transmettre, dans cet exemple : */mnt/mydata/*

Effectuer un test de transfert (ceci effectuera un test sans réaliser le transfert effectif) :

```
$ ResifDataTransfer.py \  
  --send /mnt/mydata \  
  --data-type seismic_data_miniseed --test
```

Effectuer le transfert :

```
$ ResifDataTransfer.py \  
  --send /mnt/mydata \  
  --data-type seismic_data_miniseed
```

Spécifier un fichier de configuration alternatif :

```
$ ResifDataTransfer.py \  
  --config /etc/ResifDataTransfer.conf \  
  --send /mnt/mydata \  
  --data-type seismic_data_miniseed
```

Note : lorsque le transfert *rsync* est terminé, l'opérateur obtient immédiatement un identifiant de transaction (voir 4.5).

4.4 Transmettre des métadonnées sismologiques validées

La transmission de métadonnées n'est pas implémentée dans la version actuelle.

4.5 Obtenir un identifiant de transaction, lire les sorties standards et le journal de transaction

Lors de chaque transfert de données réussi (c'est à dire : dès que le transfert *rsync* a réussi), le logiciel renvoie sur sa sortie standard (*stdout*) un identifiant de transaction. Le format de cet identifiant est décrit en Annexes.

Le logiciel émet un code de retour égal à 0 si le transfert s'est bien déroulé. Des messages d'erreurs sont parfois émis sur la sortie d'erreur standard (*stderr*).

Il appartient à l'appelant de :

- Collecter le code de retour. Si celui-ci est égal à 0 :
 - collecter sur *stdout* et conserver l'identifiant de transaction retourné
 - collecter *a posteriori* l'état de transaction (voir ci-dessous) et effectuer les actions correctives si nécessaires (exemple : ré-envoi de données initialement rejetées).
- Si celui-ci n'est pas égal à 0 :
 - collecter (ou ignorer) les messages émis sur *stderr*
 - vérifier manuellement le contenu des traces (voir l'option *log file* du fichier de

configuration)

Le logiciel renseigne un journal des transactions réussies (voir option *logbook* du fichier de configuration), qu'il est possible d'afficher de la façon suivante :

```
$ ResifDataTransfer.py \  
  --logbook \  
  [--config /etc/ResifDataTransfert.conf]
```

Le format de sortie du journal est détaillé en Annexes.

4.6 Obtenir et analyser un état de transaction

Soit la transaction identifiée de façon unique par la chaîne AAA111. Pour afficher sur la sortie standard (*stdout*) l'état de la transaction :

```
$ ResifDataTransfer.py --retrieve-logs AAA111
```

Il est possible d'effectuer un affichage structuré du XML en utilisant des outils tiers (par exemple *xmllint* ou *xmlstarlet*). Exemple :

```
$ ResifDataTransfer.py --retrieve-logs AAA111 | xmllint --format -
```

Noter qu'un état de transaction évolue dans le temps : en effet, les tâches de vérification et d'intégration de données sont insérées dans une file d'attente au niveau du centre de distribution et traitées au fur et à mesure. Le fichier XML est complété « par morceaux » (voir Annexes).

Les états de transaction sont conservés sans limite de temps par le centre de distribution.

Un utilitaire est fourni afin d'*aplatir* le contenu du fichier XML. Cet outil lit sur son entrée standard (*stdin*) un fichier XML d'état de transaction. Il est possible de le chaîner avec le logiciel de transfert, soit par un *pipe* soit par un fichier intermédiaire. Exemple :

```
$ ResifDataTransfer.py --retrieve-logs AAA111 > AAA111.xml  
$ cat AAA111.xml | ResifDataTransferTransaction.py --print-  
transaction-id AAA111
```

L'aide en ligne recense les options disponibles :

```
$ ResifDataTransferTransaction.py --help
```

```
--print-transaction-id  
    affiche l'identifiant de transaction
```

```
--print-status  
    affiche le code d'état de transaction
```

- print-datatype
affiche le type de données
- print-filelist
affiche la liste de fichiers soumis initialement par le centre de collecte et de validation
- print-process-returncode *processID*
affiche le code de retour pour le traitement identifié par *processID*
- print-process-rejectedfiles *processID*
affiche la liste des fichiers rejetés lors du traitement identifié par *processID*

Exemple :

```
$ cat AAA111.xml | ResifDataTransferTransaction.py --print-  
process-returncode T1  
0
```

```
$ cat AAA111.xml | ResifDataTransferTransaction.py --print-  
process-returncode T8  
1
```

```
$ cat AAA111.xml | ResifDataTransferTransaction.py --print-  
process-rejectedfiles T8  
2010/FR/STA1/HHE.D/FR.STA1.00.HHE.D.2010.065  
2010/FR/STA2/HHZ.D/FR.STA2.00.HHZ.D.2010.073  
2012/FR/STA2/HHZ.D/FR.STA2.00.HHZ.D.2012.100
```

Note : Cet outil est actuellement à l'état de prototype et sujet à évolutions. Se reporter aux Annexes pour les détails sur le contenu du fichier *XML*.

Annexes

Format de l'identifiant de transaction

L'identifiant est une chaîne de caractères constituée de 1 à 16 caractères. Seuls les caractères *[A-Za-z0-9]* sont utilisés. Cet identifiant est unique, y compris entre transactions issues de différents noeuds de collecte et validation.

Etat d'intégration de données validées : structure du fichier XML

Ci-dessous un extrait commenté d'un fichier XML concernant l'état d'une intégration de données.

```
<?xml version="1.0" encoding="UTF-8"?>

# Les informations sur l'état de la transaction sont encapsulées dans la balise
# « transaction ». Cette balise porte des attributs qualifiant le type
# de données, son identifiant, le noeud de collecte d'origine, et le statut
# de la transaction.
#
# Signification des statuts :
# 0  données transmises via rsync au centre de distribution
# 4  données reçues au centre de distribution et en cours
#    d'analyse par le système d'intégration
# 8  données reçues au centre de distribution et analyse par le
#    système d'intégration terminée
# 128 erreur fatale durant l'analyse des données
#

<transaction
  datatype="seismic_data_miniseed"
  id="BRDP6909"
  resifnode="TEST"
  status="2">

  # Commentaire libre inséré par la chaine d'intégration
  <comment>all file-checking steps done</comment>

  # Date de création initiale de la transaction (format ISO8601).
  # Cette date étant générée par le client, il est fortement conseillé
  # que votre horloge système soit synchronisée par NTP.
  <datecreated>2013-08-28T09:45:00Z</datecreated>

  # Date de dernière modification de l'état de transaction
  # par la chaine d'intégration (format ISO8601)
  <lastupdated>2013-08-28T10:00:11Z</lastupdated>

  # Volume total des fichiers transmis, mesuré par le client.
  # Le volume est arrondi et exprimé selon l'unité symbolisée par l'attribut
  # unit (b = octets, mb = megaoctets, gb = gigaoctets)
  <clientsize unit='gb'>18.78</clientsize>
```



```

# Liste des fichiers transmis. Chaque fichier est exprimé sous
# la forme d'un chemin relatif au répertoire source.
<filelist>
<relativepath>2010/FR/CHIF/HHE.D/FR.CHIF.00.HHE.D.2010.074</relativepath>
<relativepath>2010/FR/CHIF/HHE.D/FR.CHIF.00.HHE.D.2010.065</relativepath>
<relativepath>2010/FR/CHIF/HHE.D/FR.CHIF.00.HHE.D.2010.073</relativepath>
</filelist>

# Chaque fichier présent dans filelist (ci-dessus) est soumis à
# une série de tests (process) permettant de vérifier les
# conditions d'intégration (voir documents de référence).
# Un process est doté d'un identifiant symbolique (ici « T1 »),
# invariant et documenté ci-dessous.
#
# L'ordre d'exécution est signifié sous l'attribut « rank ».
#
# Le code de retour est indiqué sous l'attribut « returncode ».
# Ces code ne présagent pas de la recevabilité des fichiers
# traités mais permettent de suivre le fonctionnement global
# de la chaîne d'intégration. Ainsi, même si 100% de fichiers miniseed
# sont considérés comme "non admissibles" par un process donné, il est
# tout à fait possible de trouver 0 en code de retour, dans le sens où le
# traitement s'est passé normalement et peut continuer vers le
# traitement suivant.
#   0      : traitement terminé avec succès
#   1      : traitement terminé avec warning (toutefois le traitement
#           suivant peut être lancé)
#  128     : erreur fatale (les traitements suivants sont abandonnés)

<process id="T1" rank="1" returncode="0">

# Commentaire libre inséré par la chaîne d'intégration.
<comment>Lorem ipsum dolor sit amet, consectetur elit. </comment>

# Liste des fichiers non admissibles
<rejectedfiles>
<relativepath>2010/FR/CHIF/HHE.D/FR.CHIF.00.HHE.D.2010.074</relativepath>
<relativepath>2010/FR/CHIF/HHE.D/FR.CHIF.00.HHE.D.2010.065</relativepath>
</rejectedfiles>

</process>

# Suivent ensuite tous les autres traitements successifs,
# décrits ci-dessous.
<process id="T2" rank="2" returncode="0">
...
</process>
...
</transaction>

```

Les *process* dédiées à l'intégration des données validées sont les suivants (voir document de référence) :

T1 | effectue un inventaire des fichiers transmis, et détermine s'ils sont bien au format *miniseed* et contiennent bien des en-têtes *miniseed*. Utilise la commande *msi* (IRIS, v3.4.3) avec les options *-d* ou *-tg* ou *-p*. Tous les fichiers sont analysés.

T2 | vérifie la syntaxe des noms de fichiers, qui doit respecter la norme SDS

(NETWORK.STATION.LOCATION.CHANNEL.TYPE.YYYY.DDD) (cf. document de référence, page 5). Tous les fichiers qui satisfont T1 sont analysés.

T3 | Vérifie le démultiplexage complet des données dans chaque fichier (cf ref, condition A, page 8). L'unicité de fréquence d'échantillonnage dans le fichier est également vérifiée. Les noms des fichiers dans lesquels le démultiplexage n'est pas complet ou qui contiennent des segments dont les fréquences d'échantillonnage ne sont pas cohérentes avec le code du canal (selon le Manuel SEED) sont pointés. Tous les fichiers qui satisfont le test T1 sont analysés, y compris les fichiers qui ne satisfont pas T2.

T4 | Vérifie la taille des blocs, qui doit être égale à 4096 octets. Les noms des fichiers dont le contenu ne satisfait pas cette condition sont pointés (cf ref, condition E, page 8). Tous les fichiers qui satisfont T1 sont analysés, y compris les fichiers qui ne satisfont pas les T2 et T3.

T5 | Vérifie le label de qualité de chaque bloc d'un fichier, qui doit être 'D','M','Q'. Les noms des fichiers dont le contenu ne satisfait pas cette condition sont pointés (cf ref, condition D, page 8). Tous les fichiers qui satisfont T1 sont analysés, y compris les fichiers qui ne satisfont pas les T2, T3 et T4.

T6 | Vérifie la cohérence entre les quatre premiers champs du nom du fichier et le *header* de chaque bloc. Les noms des fichiers dont le contenu ne satisfait pas cette condition sont pointés (cf ref, condition A, page 8 et page 5). Tous les fichiers qui satisfont T1 sont analysés, y compris les fichiers qui ne satisfont pas les T3, T4, T5.

T7 | Vérifie la durée de la fenêtre dans le fichier correspond à la durée désignée par le nom du fichier. Les noms des fichiers dont le contenu ne satisfait pas cette condition sont pointés (cf ref, condition A, page 8 et page 5). Tous les fichiers qui satisfont T1 sont analysés, y compris les fichiers qui ne satisfont pas les T3, T4, T5, T6.

T8 | Vérifie l'encodage et la compression des données. Les noms des fichiers dont le contenu ne satisfait pas cette condition sont pointés (cf ref, condition C, page 8). Tous les fichiers qui satisfont T1 sont analysés, y compris les fichiers qui ne satisfont pas les T3, T4, T5, T6, T7

T9 | Vérifie que les métadonnées connues du centre de distribution (« noeud B ») décrivent chacun des blocs d'un fichier (cf ref, condition G, page 8). Cette condition n'est pas bloquante (les données qui ne satisfont pas cette condition sont archivées par T10). Tous les fichiers qui satisfont T1 sont analysés, y compris les fichiers qui ne satisfont pas les T3, T4, T5, T6, T7 , T8

T10 | Intègre les données. Seuls les fichiers qui satisfont les tests T1 à T9 sont intégrés. Les données sont triées et l'arborescence SDS est recrée par *dataselect* (IRIS) avec le flag `-A %Y/%n/%s/%c.D/%n.%s.%l.%c.D.%Y.%j`. Le contenu de cette arborescence est ensuite copié vers l'archive de données centralisée RESIF. Les fichiers non intégrés sont pointés.

T11 | Met à jour la base de données interne du centre de données, afin que les données soient disponibles par le webservice *fdsn-dataselect*. Les fichiers non inventoriés sont pointés. Seuls les fichiers qui satisfont les tests T1 à T10 sont inventoriés.

Etat d'intégration de métadonnées validées : structure du fichier XML

La transmission de métadonnées n'est pas implémentée dans la version actuelle.

Format du journal de transaction

Ce fichier est au format *JSON*. Le contenu et le format de ce fichier étant susceptibles d'évoluer, il est recommandé de ne pas l'interpréter directement. La commande *--logbook* permet l'affichage du journal de transaction sous forme tabulaire. Ceci décrit le format de cette sortie :

Les lignes commençant par le caractère « # » sont des commentaires libres. Les autres lignes sont des valeurs séparées par des tabulations, dans cet ordre :

1. identifiant de transaction (*str*)
2. date de transaction au format ISO8601 (*str*),
3. nom du noeud RESIF transmetteur (*str*, voir fichier de configuration),
4. type de données transmises (*str*)
5. chemin complet du répertoire transmis (*str*),
6. volume total des fichiers transmis en gigaoctets (*float*)

Un caractère « - » remplace une valeur vide ou non pertinente.

Les évolutions feront en sorte que les champs additionnels viennent à la suite de ces champs (et non en insertion ou remplacement).