



Direction technique

DT/SD/TVV

Édition 2

Date : 18/09/2013

SD-N11-0168-ML-Manuels-DifferentielShape.doc

Table des matières

1. INTRODUCTION	2
2. POINTS CLE.....	2
3. MANUEL UTILISATEUR	2
3.1. IDENTIFICATION DES OBJETS HOMOLOGUES	2
3.1.1. <i>Identification par clé</i>	3
3.1.2. <i>Identification géométrique stricte</i>	3
3.1.3. <i>Identification géométrique avec seuil (dH)</i>	3
3.2. REPRESENTATION DES DIFFERENCES GEOMETRIQUES.....	3
3.2.1. <i>Champ comment_ξ (commentaire en Z)</i>	3
3.3. DIFFERENCES SEMANTIQUES	4
3.4. LES MACRO-EVOLUTIONS.....	4
3.4.1. <i>Faussees modifications</i>	4
3.4.2. <i>Fusions</i>	4
3.4.3. <i>Scissions</i>	4
3.5. LES FICHIERS DE RESULTATS	5
3.5.1. <i>Evolutions standards</i>	5
3.5.2. <i>Evolutions complexes</i>	6
3.5.3. <i>Le fichier XML des évolutions</i>	6
3.6. LE FICHER DE PARAMETRAGE.....	9
4. INSTALLATION DE L'OUTIL	10
5. EXECUTION DU DIFFERENTIEL.....	10

Manuel du différentiel shapefile

1. Introduction

Il existe déjà un logiciel de calcul de différentiel qui est utilisé par le projet échanges mais celui-ci ne fonctionne qu'à partir de deux cartes GeoConcept.

Les deux principales raisons qui ont motivé l'élaboration de ce logiciel sont :

- Le besoin d'IGN Espace de pouvoir calculer des différentiels sur des données au format shapefile.
- La nécessité de tester le socle logiciel développé par le service des développements sur un applicatif en vraie grandeur

2. Points clé.

- Le différentiel se calcule entre deux fichiers shapefile représentant deux actualités d'un même jeu de données. Nous nommerons T1 et T2 ces deux actualités.
Le nom des fichiers ne doit contenir ni espace, ni caractères spéciaux (apostrophe, tiret, accents).
- Les multi géométries (ponctuelles et linéaires) ne sont pas supportées par le différentiel.
- Dans le cas d'une identification par clé, celles-ci doivent être uniques.
- Le différentiel s'effectue sur un modèle commun aux deux sources de données (les champs communs).
- Le différentiel est « piloté » par un fichier XML qui contient les paramètres ainsi que les métadonnées.
- Le résultat est dispatché dans 4 fichiers :
 - `<class>_diffbegin` : Shapefile qui contient les objets détruits ou modifiés dans leur état à T1.
 - `<class>_diffend` : Shapefile qui contient les objets créés ou modifiés dans leur état à T2.
 - `<class>_EvolCpx` : Shapefile contenant les « macros » évolutions (fusions, scissions et éventuellement les fausses modifications)
 - `XXX_evolution` : fichier XML qui contient l'ensemble des évolutions entre T1 et T2.

3. Manuel utilisateur

3.1. Identification des objets homologues

Il existe plusieurs manières d'identifier les objets homologues :

L'identification par clé et l'identification par similitude géométrique. A l'intérieur de cette dernière catégorie, on distingue deux possibilités : l'identification géométrique stricte et l'identification géométrique avec seuil.

Deux objets A et B ont la même géométrie si :

- Ils ont le même nombre de points.
- Pour tous les points de A de rang i , le point de B au rang i est tel que :
 $|x_A - x_B| < \text{SEUIL_IDENTITE_XY}$ et $|y_A - y_B| < \text{SEUIL_IDENTITE_XY}$ et $|z_A - z_B| < \text{SEUIL_IDENTITE_Z}$ (ces seuils sont à définir dans le fichier de paramétrage XML et **ne doivent pas être nuls**).

3.1.1. Identification par clé

- Il s'agit d'un champ du fichier DBF qui définit la clé (celle-ci doit être dans les deux fichiers shapefile). **L'unicité de cette clé est vérifiée.** Si ce n'est pas le cas le différentiel n'est pas calculé.
- Les objets à T1 et T2 qui ont la même clé seront considérés comme homologues quelle que soit leur géométrie. Si les objets sont semblables il n'y aura rien dans le différentiel sinon le résultat sera une modification géométrique et/ou sémantique.
- Un objet à T1 a pour clé C1 ; Si aucun objet à T2 n'a cette même clé, le différentiel indique qu'il y a une destruction.
- Un objet à T2 a pour clé C2 ; Si aucun objet à T1 n'a cette même clé, le différentiel indique qu'il y a une création.

3.1.2. Identification géométrique stricte

- Les objets à T1 et T2 qui ont la même géométrie seront jugés homologues. S'ils ont la même sémantique, il n'y aura rien dans le différentiel sinon le résultat sera une modification sémantique.
- Un objet à T1 n'ayant aucun objet homologue à T2 sera vu comme détruit.
- Un objet à T2 n'ayant aucun objet homologue à T1 sera vu comme créé.

3.1.3. Identification géométrique avec seuil (dH)

- On commence d'abord par identifier les objets à T1 et T2 qui ont la même géométrie.
- Ensuite, on identifie les objets à T1 et T2 dont la distance de Hausdorff (2D) est inférieure à un seuil S (paramétré dans le fichier XML) → Modification géométrique et/ou sémantique.
- Un objet à T1 n'ayant aucun homologue à moins de **dH** dans le fichier à T2 sera vu comme détruit.
- Un objet à T2 n'ayant aucun homologue à moins de **dH** dans le fichier à T1 sera vu comme créé.

3.2. Représentation des différences géométriques

Le différentiel fournit des informations sur les différences géométriques entre les deux objets à T1 et T2 qui ont été identifiés comme homologues. Ces informations se présentent sous la forme de trois attributs :

- **dXY** : Cette valeur représente la distance de Hausdorff 2D.
- **dZ** : Distance de Hausdorff 3D. Celle-ci n'est calculée que si les géométries ont un Z sur tous leurs points.
- **Inverse** : Indique si les objets à T1 et T2 ont le même sens de numérisation ou pas. Cette valeur n'est calculée que pour des objets linéaires.
- **Comment_z** : A titre indicatif, indique s'il y a une amélioration ou une dégradation du Z entre les deux états T1 et T2.

3.2.1. Champ comment_z (commentaire en Z)

Ce calcul permet de déterminer à titre indicatif si le Z s'est amélioré ou dégradé entre T1 et T2. Voici le tableau récapitulatif.

Données à T1	Données à T2	Commentaire Z
Z nulle part	Z nulle part	Sans valeur
Z nulle part	Z et NoZ mélangés	Amélioration
Z nulle part	Z partout	Amélioration
Z et NoZ mélangés	Z nulle part	Dégradation
Z et NoZ mélangés	Z et NoZ mélangés	Sans valeur
Z et NoZ mélangés	Z partout	Amélioration
Z partout	Z nulle part	Dégradation
Z partout	Z et NoZ mélangés	Dégradation
Z partout	Z partout	Amélioration

3.3. Différences sémantiques

Le calcul se fait sur le modèle commun aux deux sources de données T1 et T2 (champs communs), éventuellement filtré (voir XML de paramétrage 3.6).

3.4. Les macro-évolutions

3.4.1. Fausses modifications

Il s'agit d'une destruction suivie d'une création d'un objet semblable à un seuil près (ce seuil est défini dans le fichier de paramètres par la balise <FALSE_MODIF> dans TOLERANCES).

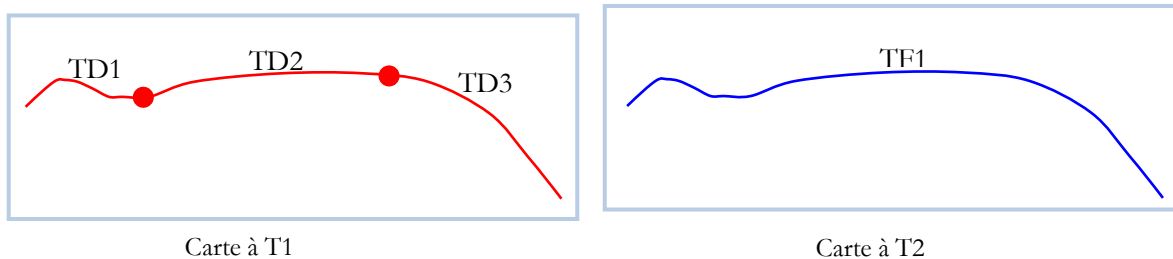
3.4.2. Fusions

Ce calcul n'est effectué pour l'instant que pour les objets linéaires.

On considère qu'il y a une fusion dans les deux cas suivants :

- Modification géométrique d'un objet et N destructions dans le cas de l'identification par clé.
- Une création et N destructions pour tous types d'identification.

Prenons le cas suivant :



Dans une identification par clé, il est possible que **TD1** ait la même clé que **TF1** ; On considère **TD1** et **TF1** « semblables » d'où une modification et deux destructions.

Par contre dans le cas d'une identification géométrique, il n'y a aucun objet de la carte de fin « semblable » à un objet de la carte de début, ce qui implique que l'on trouve trois destructions et une création.

Elles sont identifiées de la manière suivante :

- Pour l'objet **TF1** de <class>_diffEnd, on recherche s'il existe des objets de <class>_diffBegin qui pourraient en constituer une partie. On calcule pour cela les demi Hausdorff entre **TD1** et **TF1**, **TD2** et **TF1** et **TD3** et **TF1**.
- Si ces distances sont inférieures à un certain seuil (balise MERGE_SPLIT du fichier de paramétrage), elles sont candidates à la fusion.
- On effectue ensuite la fusion de **TD1**, **TD2** et **TD3** et on regarde si la distance de Hausdorff entre cet objet résultant et **TF1** est inférieure au seuil ; Si c'est le cas, on considère que l'on a trouvé une fusion.

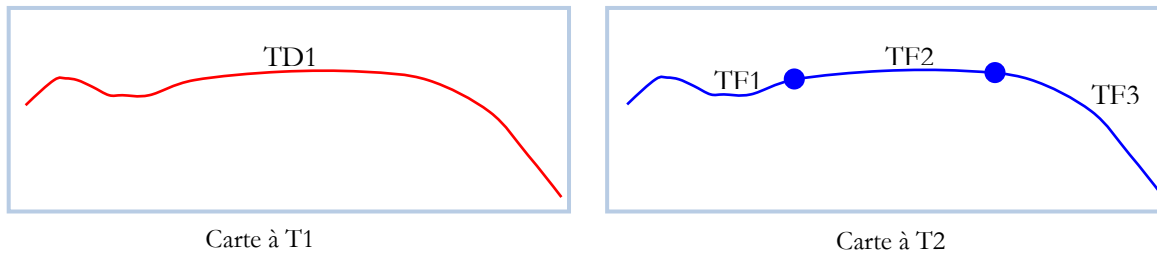
3.4.3. Scissions

Ce calcul n'est effectué pour l'instant que pour les objets linéaires.

On considère qu'il y a une scission dans les deux cas suivants :

- Modification géométrique d'un objet et N créations dans le cas de l'identification par clé.
- Une destruction et N créations pour tous types d'identification.

Prenons le cas suivant :



Dans une identification par clé, il est possible que **TD1** ait la même clé que **TF1** ; On considère **TD1** et **TF1** « semblables » d'où une modification et deux créations. Par contre dans le cas d'une identification géométrique, il n'y a aucun objet de la carte de fin « semblable » à un objet de la carte de début, ce qui implique que l'on trouve une destruction et trois créations.

Elles sont identifiées de la manière suivante :

- Pour l'objet **TD1** de `<class>_diffBegin`, on recherche s'il existe des objets de `<class>_diffEnd` qui pourraient en être une partie. On calcule pour cela les demi Hausdorff entre **TF1** et **TD1**, **TF2** et **TD1** et **TF3** et **TD1**.
- Si ces distances sont inférieures à un certain seuil (balise `MERGE_SPLIT` du fichier de paramétrage), elles sont candidates à la scission.
- On effectue ensuite la fusion de **TF1**, **TF2** et **TF3** et on regarde si la distance de Hausdorff entre cet objet résultant et **TD1** est inférieure au seuil ; Si c'est le cas, on considère que l'on a trouvé une scission.

3.5. Les fichiers de résultats

3.5.1. Evolutions standards

En plus du modèle commun aux deux sources de données, on ajoute des informations supplémentaires dans les fichiers de résultats. [x] indique si ce champ est renseigné pour les différents types d'évolutions.

En bleu les champs communs aux deux fichiers `<class>_diffbegin` et `<class>_diffend`.

En rouge les champs qui sont uniquement dans `<class>_diffend`.

Champs	Définition	Création	Suppression	Modification	Fusion	Scission	Fausse modif.
<code>_key</code> (chaîne)	Identifiant de l'objet	x	x	x	x	x	x
<code>_evol_type</code> (chaîne)	Type d'évolution (creation, deletion ou modification)	x	x	x	x	x	x
<code>_geo_modif</code> (booléen)	La modification est-elle géométrique ?			x	x	x	x
<code>_sem_modif</code> (booléen)	La modification est-elle sémantique ?			x	x	x	x
<code>_dXY</code> (double)	Distance de Hausdorff 2D			x			x
<code>_dZ</code> (double)	Distance de Hausdorff 3D			x			x
<code>_inverse</code> (booléen)	Indique si les objets ont le même sens de numérisation ou pas. Cette valeur n'est calculée que pour des objets linéaires			x			x
<code>_comment_z</code> (chaîne)	Indique s'il y a une amélioration ou une dégradation du Z entre l'objet à T1 et T2.			x			x

3.5.2. Evolutions complexes

Ce fichier contient des objets surfaciques englobant un phénomène tel qu'une fusion, scission ou fausse modification. Sa structure est la suivante :

Champs	Définition
_evol_type	Type d'évolution (merge, split et false modification)
_geo_modif (booléen)	Pour les fusions et scissions indique si un objet de départ a été modifié géométriquement ou pas.
_sem_modif (booléen)	Modification sémantique entre les objets à T1 et T2 (fusions, scissions)
_dXY	Voir tableau précédent
_dZ	Voir tableau précédent
_inverse	Voir tableau précédent
_comment_z	Voir tableau précédent
_lk_begin	Rangs des objets dans le fichier shapefile diffbegin séparés par '/'
_lk_end	Rangs des objets dans le fichier shapefile diffend séparés par '/'

Les champs en rouge ne sont renseignés que pour les fausses modifications.

3.5.3. Le fichier XML des évolutions

Son nom dépend du format décrit dans le fichier de paramètres. La géométrie est dans le format gml.

Ce format ne sera pas décrit ici.

Contenu de la balise METADATA

<pre><VERSION>01_05_00</VERSION> <DATE>04/10/2010</DATE> <DATA_T1> D:\USERS\T1\ROUTE.shp</ DATA_T1> < DATA_T2> D:\USERS\T2\ROUTE.shp </ DATA_T2> <DUREE>1 minute(s) et31 seconde(s)</DUREE> <NB_OBJETS ini= »615831 » fin= »613719 »/></pre>	<ul style="list-style-type: none"> ▪ Version du logiciel utilisé ▪ Date du calcul ▪ Données à T1 ▪ Données à T2 ▪ Durée du calcul ▪ Nombre d'objets ini(à T1) et fin(à T2)
<pre><id_column> <geometry_column></pre>	(voir 3.6) Concerne uniquement le différentiel PostgreSQL
Identification	(voir 3.6)
Tolérances	(voir 3.6)
<pre>Métadonnées <CHAMP id="dest" nom="Destinataire">CG 77</CHAMP> <CHAMP id="prod" nom="Produit">BDT</CHAMP> <CHAMP id="proj" nom="Projection">L93</CHAMP> ...</pre>	Métadonnées (définies par l'utilisateur)

Balise MAJ_OBJ

Type d'évolution	Description dans le fichier
Destruction (deletion)	<pre><MAJ_OBJ type="D"> <EVOL_OBJ etat="OLD" type="D" cleabs="XXXX"> </MAJ_OBJ></pre>
Création (creation)	<pre><MAJ_OBJ type="C"> <EVOL_OBJ etat="NEW" type="C" cleabs="XXXX"> </MAJ_OBJ></pre>
Modification géométrique	<pre><MAJ_OBJ type="MG"> <MESURE> <DHP>xxx</DHP> <DHZ>xxx</DHZ> <Inversion></Inversion> </MESURE> <EVOL_OBJ etat="OLD" type="MG" cleabs="XXXX"> <EVOL_ELEM_GEOM> </EVOL_ELEM_GEOM> </EVOL_OBJ> <EVOL_OBJ etat="NEW" type="MG" cleabs="XXXX"> <EVOL_ELEM_GEOM> </EVOL_ELEM_GEOM> </EVOL_OBJ> </MAJ_OBJ></pre>
Modification sémantique	<pre><MAJ_OBJ type="MS"> <EVOL_OBJ etat="OLD" type="MS" cleabs="XXXX"> <EVOL_ELEM_ATT name="IT_VERT" value="NC"> </EVOL_ELEM_ATT> <EVOL_ELEM_ATT name="NOM_RUE_D" value="NR"> </EVOL_ELEM_ATT> </EVOL_OBJ> <EVOL_OBJ etat="NEW" type="MG" cleabs="XXXX"> <EVOL_ELEM_ATT name="IT_VERT" value=" "> </EVOL_ELEM_ATT> <EVOL_ELEM_ATT name="NOM_RUE_D" value=" "> </EVOL_ELEM_ATT> </EVOL_OBJ> </MAJ_OBJ></pre>
Modification mixte (géométrie + sémantique)	<pre><MAJ_OBJ type="MG"> <MESURE> <DHP>xxx</DHP> <DHZ>xxx</DHZ> <Inversion></Inversion> </MESURE> <EVOL_OBJ etat="OLD" type="MG" cleabs="XXXX"> <EVOL_ELEM_ATT name="IT_VERT" value="NC"> </EVOL_ELEM_ATT> <EVOL_ELEM_ATT name="NOM_RUE_D" value="NR"> </EVOL_ELEM_ATT> </EVOL_OBJ></pre>

	<pre> <EVOL_ELEM_GEOM> </EVOL_ELEM_GEOM> </EVOL_OBJ> <EVOL_OBJ etat="NEW" type="MG" cleabs="XXXX"> <EVOL_ELEM_ATT name="IT_VERT" value=" "> </EVOL_ELEM_ATT> <EVOL_ELEM_ATT name="NOM_RUE_D" value=" "> </EVOL_ELEM_ATT> </EVOL_ELEM_GEOM> </EVOL_OBJ> </MAJ_OBJ> </pre>
Fausse modification	<pre> <MAJ_OBJ type="FM"> <MESURE> <DHP>xxx</DHP> <DHZ>xxx</DHZ> <Inversion></Inversion> </MESURE> <EVOL_OBJ etat="OLD" type="D" cleabs="XXXX"> </EVOL_OBJ> <EVOL_OBJ etat="NEW" type="C" cleabs="XXXX"> </EVOL_OBJ> </MAJ_OBJ> </pre>
Fusion (merge)	<pre> <MAJ_OBJ type="F"> <EVOL_OBJ etat="OLD" type="MM" cleabs="cle1"> <EVOL_ELEM_ATT name="Nom" value="TOTO"> </EVOL_ELEM_ATT> </EVOL_OBJ> <EVOL_OBJ etat="OLD" type="D" cleabs="cle2"> </EVOL_OBJ> <EVOL_OBJ etat="NEW" type="MM" cleabs="cle1"> <EVOL_ELEM_ATT name="Nom" value="TOTO"> </EVOL_ELEM_ATT> </EVOL_OBJ> </MAJ_OBJ> </pre>
Scission (split)	<pre> <MAJ_OBJ type="S"> <EVOL_OBJ etat="OLD" type="MG" cleabs="cle1"> </EVOL_OBJ> <EVOL_OBJ etat="NEW" type="MG" cleabs="cle1"> </EVOL_OBJ> <EVOL_OBJ etat="NEW" type="C" cleabs="cle2"> </EVOL_OBJ> </MAJ_OBJ> </pre>

3.6. Le fichier de paramétrage

Il commence avec la balise **METADATA**

<p><METADATA></p> <p><code><id_column>ID</id_column></code></p>	<p>Balise principale</p> <p>Le nom de la colonne définissant l'identificateur</p> <ul style="list-style-type: none"> • Cette balise n'est pas obligatoire (par défaut l'id sera le rang dans le fichier) • Elle peut être vide (par défaut l'id sera le rang dans le fichier)
<p><IDENTIFICATION> <code><TYPE>CLE</TYPE></code> </IDENTIFICATION></p> <p>Les autres types d'identification ainsi que leurs balises associées.</p> <ul style="list-style-type: none"> • Identification géométrique (stricte) <code><TYPE>GEOM</TYPE></code> • Identification géométrique avec seuil <code><TYPE>SEUIL</TYPE></code> <code><SEUIL>2</SEUIL></code> (valeur en mètres) 	<p style="text-align: center;">Identification</p> <p>Type d'identification, ici :</p> <ul style="list-style-type: none"> • Par clé <p>Distance de Hausdorff inférieure ou égale à 2 mètres</p>
<p><TOLERANCES> <code><MERGE_SPLIT>1</ MERGE_SPLIT ></code> <code><FALSE_MODIF>1</ FALSE _MODIF></code></p> <p><code><PRECISION>2</PRECISION></code></p> <p><code><SEUIL_EGALITE_XY>0.01</ SEUIL_EGALITE_XY></code> <code>< SEUIL_EGALITE_Z>0.01</ SEUIL_EGALITE_Z></code></p> <p></TOLERANCES></p>	<p style="text-align: center;">Tolérances pour le calcul (degrés ou mètres).</p> <ul style="list-style-type: none"> • Fusions/scissions (Pas obligatoire, par défaut 1.) • Fausses modifications scissions (Pas obligatoire et utilisée uniquement dans le cas d'une identification par clé, par défaut 1.). • Précision pour l'écriture des résultats : 2 signifie deux chiffres après la virgule (cette valeur doit être entre 0 et 7). Cette balise n'est pas non plus obligatoire (par défaut 2) • Seuils pour définir que deux points sont égaux $x-x' < SEUIL_EGALITE_XY$ $y-y' < SEUIL_EGALITE_XY$ et $z-z' < SEUIL_EGALITE_Z$
<p><code><CHAMP id="dest" nom="Destinataire">CG 77</CHAMP></code> <code><CHAMP id="prod" nom="Produit">BDT</CHAMP></code> <code><CHAMP id="proj" nom="Projection">L93</CHAMP></code> <code><CHAMP id="archi1" nom="Archivage initial">093</CHAMP></code> <code><CHAMP id="archi2" nom="Archivage final">102</CHAMP></code></p>	<p style="text-align: center;">Métadonnées</p> <p>Ces champs ont pour attributs :</p> <ul style="list-style-type: none"> • Un identifiant (id) qui sert pour le format (voir plus bas) • Un nom.
<p><code><FORMAT sep="_" suffixe="evolutions"></code> <code><ID_PART>dest</ID_PART></code> <code><ID_PART>prod</ID_PART></code> <code><ID_PART>proj</ID_PART></code> <code><ID_PART>archi1</ID_PART></code> <code><ID_PART>archi2</ID_PART></code> <code><ID_PART>emprise</ID_PART></code> </FORMAT></p>	<p style="text-align: center;">Format du fichier XML des évolutions en sortie</p> <p>Les balises ID_PART font références aux identifiants des champs de saisie. Dans cet exemple, le nom du fichier sera [dest]_[prod]_[proj]_[archi1]_[archi2]_[theme]_evolutions.xml soit :</p> <p>CG 77_BDT_L93_093_102_evolutions.xml</p> <p>S'il n'y a aucune balise ID_PART ou si celles-ci sont vides le nom du fichier sera evolutions.xml.</p>

<pre><MODEL> <COLUMN>nom (DBF ou colonne potsgre)</COLUMN> ... </MODEL> </METADATA></pre>	<h3 style="background-color: #4F81BD; color: white; padding: 5px;">Filtrage des colonnes</h3> <p>Cette balise permet de filtrer les attributs que l'on veut conserver pour le calcul du différentiel.</p> <p>Important : la balise définissant la colonne clé (id_column) doit être présente dans la balise MODEL.</p>
--	---

4. Installation de l'outil

- Via le site du service des développements (sd.ign.fr)
 - Rubrique TVV (vecteur)/Logiciels de l'unité.
 - Aller dans le dossier Différentiel shapefile → Les livraisons
 - Télécharger la version pour MAC ou Windows.

5. Exécution du différentiel

- Créer un fichier de paramètres XML (voir 3.1)
- Lancer la commande **differentiel_shape** avec les paramètres suivants :
 - v mode verbeux
 - params <config file>.xml
 - file_t1 <shapefile à T1>
 - file_t2 <shapefile à T2>
 - class <nom de la classe>
 - output <répertoire de sortie des résultats> *Optionnel (par défaut le répertoire courant)*

FIN DU DOCUMENT