



## **1Water**

---

# **Guide Administrateur Version 3.4**

08/07/2025

---

## A propos du Guide

---

Les informations contenues dans ce document pourront faire l'objet de modifications sans préavis de la part de 1Spatial.

Sauf mention contraire, les sociétés, les noms et les données utilisés dans les exemples sont fictifs.

Aucune partie de ce document ne peut être reproduite ou transmise à quelque fin ou par quelque moyen que ce soit, électronique ou mécanique, sans la permission expresse et écrite de 1Spatial.

---

## Limitation de garantie et de responsabilité

---

1Spatial a réalisé tous les tests nécessaires et suffisants du Progiciel et a vérifié la conformité de son guide.

Le client reconnaît que dans l'état actuel de la technologie informatique, le fonctionnement du Progiciel est susceptible d'être interrompu ou affecté par des bogues. 1Spatial ne garantit pas que le Progiciel, le média d'installation, la clef ou la documentation livrés soient exempts d'erreurs, de bogues ou d'imperfections.

Ainsi, le client doit effectuer toutes les sauvegardes, prévoir et effectuer toutes les procédures de remplacement en vue d'éventuelles défaillances du Progiciel, prendre toutes les mesures appropriées pour se prémunir contre toute conséquence dommageable due à l'utilisation ou la non-utilisation du Progiciel.

Il est expressément convenu que 1Spatial ne sera en aucun cas responsable des dommages directs ou indirects dus à l'utilisation du Progiciel.

L'utilisation du Progiciel est soumise à la signature par le client du contrat de droit d'utilisation des logiciels de 1Spatial.

---

## Marques déposées

---

**Ce progiciel** est une marque déposée de 1Spatial.

**Ce progiciel**, développé par 1Spatial, est une marque déposée et est la propriété exclusive de 1Spatial. Toutes les autres marques citées appartiennent à leurs propriétaires respectifs.

Copyright © 2025, 1Spatial. Tous droits réservés.

## TABLE DES MATIÈRES

<b>1</b>	<b>INTRODUCTION .....</b>	<b>10</b>
<b>2</b>	<b>CONFIGURATION GÉNÉRALE.....</b>	<b>11</b>
2.1	Licences.....	11
2.2	Portails .....	11
2.3	Serveurs .....	11
2.4	Utility Network.....	12
2.5	Réseau.....	12
<b>3</b>	<b>CONFIGURATION DES APPLICATIONS .....</b>	<b>13</b>
3.1	Eau potable .....	13
3.1.1	General.....	13
3.1.2	Anomalie .....	21
3.1.3	Coupure d'eau.....	23
3.1.4	Export de données .....	28
3.1.5	Recherche de fuite .....	29
3.1.6	Profil en long .....	29
3.1.7	Notation .....	30
3.1.8	Mise à jour.....	32
3.1.9	Cadastre .....	35
3.1.10	Gestion des conséquences .....	37
3.2	Assainissement .....	38
3.2.1	General.....	38
3.2.2	Anomalie .....	47
3.2.3	Encrassement.....	49
3.2.4	Point noir.....	52
3.2.5	Export de données .....	54
3.2.6	Propagation .....	54
3.2.7	Profil en long .....	55
3.2.8	Inspection télévisée .....	56
3.2.9	Intervention.....	66
3.2.10	Notation .....	70
3.2.11	Mise à jour.....	72
3.2.12	Cadastre .....	75
3.2.13	Gestion des conséquences .....	77
<b>4</b>	<b>CONFIGURATION DE LA MOBILITÉ.....</b>	<b>79</b>
4.1	Authentification OAuth/SSO.....	79
4.2	Fichiers de configuration sur le mobile.....	80
4.2.1	Fichier « applicationConfig.json » .....	80
4.2.2	Fichier « logConfig.json » .....	82
4.2.3	Fichier « userConfig.json » .....	82
4.2.4	Gestion de la sécurité.....	83

---

4.3	Fichiers de configuration sur le serveur.....	83
4.4	Gestion des filtres .....	83
4.5	Condition Simple.....	84
4.5.1	Champs obligatoires.....	84
4.5.2	Champs optionnels.....	85
4.5.3	Mots-clés.....	86
4.5.4	Méthodes disponibles .....	86
4.5.5	Exemples .....	86
4.6	Filtre simple .....	87
4.6.1	Champs obligatoires.....	87
4.6.2	Exemple.....	87
4.7	Filtre complexe .....	88
4.7.1	Champs obligatoires.....	88
4.7.2	Exemple.....	88
4.8	Configuration de la recherche par adresse.....	89
4.9	Configuration du module « Graffiti ».....	90
4.10	Configuration de la fonctionnalité « Gestion des conséquences » .....	90
<b>5</b>	<b>UTILITAIRES.....</b>	<b>92</b>
5.1	Export EPANET .....	92
5.1.1	Fichier de paramétrage .....	92
5.1.2	Exemple.....	100
5.2	Export SWMM.....	100
5.2.1	Fichier de paramétrage .....	101
5.2.2	Exemple.....	108
5.3	Export StaR-DT .....	109
5.3.1	Définition.....	110
5.3.2	Exporter les données du réseau au format StaR-DT .....	110
5.3.3	Modèle conceptuel de données.....	110
5.3.4	Correspondance 1Water .....	111
5.3.5	Fichier de paramétrage .....	111
5.4	Méthode de notation.....	117
5.4.1	Note.....	117
5.4.2	Critère .....	119
<b>6</b>	<b>ANNEXES.....</b>	<b>133</b>
6.1	Filtres de sélection personnalisés .....	133
6.1.1	Généralités .....	133
6.1.2	Assainissement.....	133
6.1.3	Eau.....	135
6.2	Définir les cartes .....	136

---

---

6.3	Gestion de la visibilité des associations de contenance .....	140
6.4	Profil en long .....	141
6.4.1	Configuration de la trace .....	141
6.4.2	Configuration du rapport .....	141

## FIGURES

Figure 1 : Informations sur le réseau de distribution (Eau).....	14
Figure 2 : Localisation (Eau).....	14
Figure 3 : Localisation – Restriction de la zone de recherche (Eau).....	14
Figure 4 : Impression (Eau).....	16
Figure 5 : Couches d'exploitation (Eau).....	17
Figure 6 : Couches de sectorisation (Eau) .....	18
Figure 7 : Couches de dépose (Eau) .....	18
Figure 8 : Rapports (Eau).....	19
Figure 9: Service d'impression (Eau) .....	19
Figure 10 : Modèle numérique de terrain (Eau).....	21
Figure 11 : Anomalie (Eau) .....	22
Figure 12 : Patrimoine pour anomalie (Eau) .....	22
Figure 13 : Coupure - Trace .....	25
Figure 14 : Coupure - Filtres .....	25
Figure 15 : Coupure - Éléments impactés .....	27
Figure 16 : Export (Eau) .....	29
Figure 17 : Recherche de fuites .....	29
Figure 18 : Notation (Eau) .....	30
Figure 19 : Définition des notes (Eau) .....	32
Figure 20 : Prototype « Point de livraison » .....	33
Figure 21 : Configuration du cadastre sous PostgreSQL (Eau) .....	35
Figure 22 : Informations sur le réseau de distribution (Assainissement) .....	38
Figure 23 : Localisation (Assainissement).....	39
Figure 24 : Localisation - Restriction de la zone de recherche (Assainissement).....	39
Figure 25 : Impression (Assainissement).....	40
Figure 26 : Couches d'exploitation (Assainissement).....	41
Figure 27 : Couches de sectorisation (Assainissement) .....	43
Figure 28 : Rapports (Assainissement) .....	44
Figure 29 : Service d'impression (Assainissement) .....	45
Figure 30 : Modèle numérique de terrain (Assainissement).....	47
Figure 31 : Anomalie (Assainissement) .....	47
Figure 32 : Patrimoine pour anomalie (Assainissement) .....	48
Figure 33 : Encrassement .....	50
Figure 34 : Point noir .....	52
Figure 35 : Export (Assainissement) .....	54
Figure 36 : Propagation (Assainissement 2 domaines) .....	55
Figure 37 : Propagation (Assainissement 1 domaine).....	55
Figure 38 : Généralités ITV .....	56
Figure 39 : Patrimoine pour les inspections du réseau .....	57
Figure 40 : Filtres attributaires pour les inspections du réseau.....	58

---

Figure 41 : Trace pour les inspections du réseau .....	59
Figure 42 : Exemples de conséquence ITV .....	60
Figure 43 : Outil de génération du plan ITV .....	61
Figure 44 : Généralités pour intervention (Assainissement).....	66
Figure 45 : Patrimoine pour les interventions (Assainissement) .....	67
Figure 46 : Filtres attributaires pour les interventions (Assainissement) .....	68
Figure 47 : Trace pour les interventions (Assainissement) .....	69
Figure 48 : Patrimoine pour la notation (Assainissement).....	70
Figure 49 : Définition des notes (Assainissement) .....	71
Figure 50 : Prototype « Point de collecte (eaux usées) » .....	73
Figure 51 : Configuration du cadastre sous PostgreSQL (Assainissement) .....	75
Figure 52 : Panneau d'enregistrement d'une application.....	79
Figure 53 : Application enregistrée avec son identifiant .....	80
<b>Figure 115 : Problème de certificat de sécurité</b> .....	82
Figure 54 : Configuration de la gestion des conséquences en mobilité.....	90
Figure 55 : Mise en évidence d'une conséquence .....	91
Figure 56 : Partie « headers ».....	93
Figure 57 : Partie « options » .....	93
Figure 58 : Partie "domains" .....	93
Figure 59 : Entité d'une section.....	94
Figure 60 : Section "SYSTEM" .....	94
Figure 61 : Filtre sur une entité .....	94
Figure 62 : Valeur avec dictionnaire et « next ».....	95
Figure 63 : Valeur opération de comparaison et plusieurs « next » .....	95
Figure 64 : Exemple de dictionnaire avec un opérateur de comparaison d'égalité implicite.....	96
Figure 65 : Exemple de dictionnaire avec un opérateur de comparaison strictement inférieur .....	96
Figure 66 : Exemple d'un modèle d'attributs .....	97
Figure 67 : Exemple d'un modèle conditionnel d'attributs.....	98
Figure 68 : Partie « Paramètres ».....	99
Figure 69 : Exemple de la section « PIPES » .....	100
Figure 70 : Partie « headers ».....	101
Figure 71 : Partie "options" .....	102
Figure 72 : Partie "domains" .....	102
Figure 73 : Entité d'une section.....	103
Figure 74 : Section "SYSTEM" .....	103
Figure 75 : Filtre sur une entité .....	103
Figure 76 : Valeur avec dictionnaire et « next ».....	104
Figure 77 : Valeur opération de comparaison et plusieurs « next » .....	104
Figure 78 : Exemple de dictionnaire avec un opérateur de comparaison d'égalité implicite.....	105
Figure 79 : Exemple de dictionnaire avec un opérateur de comparaison strictement inférieur .....	105
Figure 80 : Exemple d'un modèle d'attributs .....	106
Figure 81 : Exemple d'un modèle conditionnel d'attributs.....	107
Figure 82 : Partie « Paramètres » .....	108

---



Figure 83 : Exemple de la section « OUTFALLS ».....	108
Figure 84 : Exemple de modèle d'attributs « OUTFALLS ».....	109
Figure 85 : Partie « headers ».....	112
Figure 86 : Contenu de « headers » .....	112
Figure 87 : Partie « Paramètres ».....	112
Figure 88 : Exemple de dictionnaire avec un opérateur de comparaison d'égalité implicite.....	113
Figure 89 : Exemple de dictionnaire avec un opérateur de comparaison supérieur ou égal .....	113
Figure 90 : Partie "domains" .....	114
Figure 91 : Entité d'une section.....	114
Figure 92 : Filtre sur une entité .....	114
Figure 93 : Valeur avec dictionnaire et « next ».....	115
Figure 94 : Exemple d'un modèle d'attributs .....	116
Figure 95 : Concaténation de 2 valeurs.....	117
Figure 96 : Rechercher une valeur dans un dictionnaire .....	117
Figure 97 : Configuration de la trace pour le profil en long .....	141
Figure 98 : Paramètres généraux du profil en long.....	142
Figure 99 : Tableau d'information du profil en long .....	142
Figure 100 : Définition du tableau d'informations du profil en long .....	142
Figure 101 : Balises obligatoires d'une information complémentaire .....	143
Figure 102 : Formatage numérique d'une information complémentaire .....	143
Figure 103 : Formatage date d'une information complémentaire .....	144
Figure 104 : Concaténation de champs dans une information complémentaire .....	144
Figure 105 : Tracé du profil en long.....	145
Figure 106 : Définition des couleurs du profil en long .....	145
Figure 107 : Expression d'un calcul au niveau du tracé .....	146
Figure 108 : Libellés sur un profil en long .....	147
Figure 109 : Définition d'un libellé du profil en long.....	147
Figure 110 : Définition d'un symbole du profil en long.....	148
Figure 111 : Paramétrage du tracé des linéaires pour le profil.....	148
Figure 112 : Couleur du tracé de la pente du profil .....	148
Figure 113 : Couleur de tracé de la contre-pente du profil.....	148
Figure 114 : Paramétrage pour récupérer la cote fil d'eau des linéaires.....	149
Figure 115 : Tracé d'un linéaire utilisant la cote radier des regards .....	150
Figure 116 : Tracé d'un linéaire utilisant ses côtés fil d'eau.....	150
Figure 117 : Paramétrage pour récupérer la dimension des linéaires .....	151
Figure 118 : Paramétrage pour récupérer la longueur des linéaires .....	151
Figure 119 : Paramétrage du tracé du terrain.....	152
Figure 120 : Exemple de tracé du terrain basé sur le tampon des regards .....	152
Figure 121 : Paramétrage pour récupérer la côté de surface à partir des ponctuels .....	153
Figure 122 : Paramétrage pour récupérer la côté de surface au niveau du premier point .....	154
Figure 123 : Paramétrage pour récupérer la côté de surface au niveau du dernier point .....	154
Figure 124 : Exemple tracé des ponctuels en extrémité.....	155
Figure 125 : Paramétrage du tracé des ponctuels en extrémité.....	155



---

Figure 126 : Paramétrage du tracé des ponctuels d'extrémité (Eau) .....	156
Figure 127 : Exemple d'un tracé de ponctuels d'extrémité (Eau) .....	156
Figure 128 : Paramétrage du tracé des ponctuels d'extrémité (Assainissement) .....	157
Figure 129 : Paramétrage de la partie radier du regard (Assainissement) .....	157
Figure 130 : Paramétrage de la partie tampon du regard (Assainissement) .....	158
Figure 131 : Exemple de tracé d'un regard (Assainissement) .....	158
Figure 132 : Paramétrage du tracé des ponctuels en vertex .....	159
Figure 133 : Exemple d'info-bulle au niveau d'un raccord de branchement .....	159
Figure 134 : Paramétrage MNT du module « Profil en long » .....	159

# 1 INTRODUCTION

---

Ce document décrit tous les paramètres de configuration de l'application 1Water.

Les différents niveaux de configuration suivants sont décrits :

- ▶ **Serveur** : noms des licences, portails et serveurs et configuration d'Utility Network ;
- ▶ **Applications** : eau potable et assainissement ;
- ▶ **Mobilité** : filtres, recherche par adresse, ... ;
- ▶ **Utilitaires** : export EPANET, SWMM, StaR-DT et méthode de notation ;
- ▶ **Annexes** : exemples de filtres personnalisés et description des JSON pour la définition des cartes.

## 2 CONFIGURATION GÉNÉRALE

---

1Water est hébergé dans le serveur 1BiZ Server, la configuration générale du serveur est donc celle du serveur « 1BiZ Server ». Elle est décrite dans le document « **1BiZ Server Guide Administrateur** ».

Ce chapitre apporte des précisions concernant la configuration de « 1Water » dans la configuration générale de « 1BiZ Server ».

### 2.1 LICENCES

Le nom du produit à renseigner dans « name » de « products » du fichier « **modules.json** » est « **1Water** ».

Le produit « **1Water** » propose deux modules : « **water** » et « **sanitation** ».

### 2.2 PORTAILS

Un portail doit être déclaré pour « 1Water », soit sous le nom « default » soit sous un nom dédié (ex : « 1water-portal »), dans le fichier « **arcgis-connector.json** ».

### 2.3 SERVEURS

Deux serveurs doivent être déclarés dans le fichier « **arcgis-connector.json** ».

Vous pouvez utiliser le nom par défaut (« default » et « default-service ») ou personnaliser le nom (ex : « 1water-arcgis » et « 1water-arcgis-service »).

Le nom du serveur « default », ou le nom personnalisé, est utilisé dans les fichiers « config.json » de chacun des métiers (« **onewater\sanitation\config.json** » et « **onewater\water\config.json** »). Dans la suite du document, le nom « default » est utilisé.

Le nom du serveur « default-service » ou le nom personnalisé, est utilisé dans le fichier « **onebizserver.properties** » pour chacun des métiers (« 1water.sanitation.server » et « 1water.water.server »).

Lors du démarrage du serveur onebiz, il peut y avoir un problème de connexion vers les serveurs déclarés dans le fichier « **arcgis-connector.json** ». Un message « [operation=initialize()][TAG=CONNECTION] Could not get token ... » apparaît alors dans le log. Il y a alors 2 possibilités :

- ▶ soit il y a une erreur de configuration dans le fichier « **arcgis-connector.json** », et alors le message dans le log l'indique
- ▶ soit il n'y a pas d'erreur de configuration dans le fichier « **arcgis-connector.json** » et alors ce message veut dire que le démarrage du serveur déclaré dans le fichier « **arcgis-connector.json** » est lent. Dans ce cas il faut augmenter la valeur de la ressource « **application.connection.retrycount** » dans le fichier « **onebiz.properties** ».

---

## 2.4 UTILITY NETWORK

1Water nécessite la configuration d'un réseau de distribution pour le métier « Eau potable » et d'un réseau de distribution pour le métier « Assainissement ». Ces configurations sont définies dans le fichier « **arcgis-utilitynetworks.json** ».

## 2.5 RÉSEAU

Si votre organisation filtre les domaines autorisés, pour que les fonctionnalités de 1Water Web fonctionnent correctement, les éléments suivants sont requis :

- ▶ [fonts.googleapi.com](https://fonts.googleapis.com)
- ▶ [fonts.gstatic.com](https://fonts.gstatic.com)
- ▶ [js.arcgis.com](https://js.arcgis.com)

## 3 CONFIGURATION DES APPLICATIONS

---

Ce chapitre donne des précisions sur la configuration des applications « Eau potable » et « Assainissement » de 1Water via des fichiers json.

Certains modules sont configurables, entièrement ou en partie, à l'aide d'une application de configuration dédiée (voir les chapitres « Configuration » de chaque application dans le guide utilisateur de 1Water pour plus d'informations).

Les modules gérés dans l'application de configuration « Eau potable » sont :

- ▶ « Anomalie » : les filtres personnalisés (filters) ne sont pas gérés, pour le moment, dans l'application de configuration ;
- ▶ « Recherche de fuite » : la valeur de tolérance d'accroche (buffer) n'est pas gérée, pour le moment, dans l'application de configuration.

Les modules gérés dans l'application de configuration « Assainissement » sont :

- ▶ « Anomalie » : les filtres personnalisés (filters) ne sont pas gérés, pour le moment, dans l'application de configuration ;
- ▶ « Encrassement » : les filtres personnalisés (filters) ne sont pas gérés, pour le moment, dans l'application de configuration ;
- ▶ « Point Noir » : les filtres personnalisés (filters) ne sont pas gérés, pour le moment, dans l'application de configuration ;
- ▶ « Intervention » (« Curage » et « Visite ») : plusieurs options de configuration ne sont pas gérées, pour le moment, dans l'application de configuration :
  - la distance en m (rayon) pour construire la géométrie du programme d'intervention (buffer) ;
  - les filtres personnalisés (filters) ;
  - les nœuds en extrémité des linéaires inspectés et non pris en charge par l'intervention (linkedNodes) ;
  - la configuration de trace (traceConfigurations).
- ▶ « Propagation » : géré entière par l'application de configuration.

! Aucun changement, au niveau des fichiers json, n'est pris en compte si l'interface de l'application de configuration le permet. Dans ce cas, les fichiers json sont utilisés uniquement pour initialiser les valeurs par défaut au niveau de l'interface. Pour les propriétés qui ne sont pas encore gérées par l'interface, les modifications restent au niveau des fichiers json.

### 3.1 EAU POTABLE

#### 3.1.1 General

La configuration générale de l'application « Eau potable » est définie dans le fichier « `\config\onewater\water\config.json` »

### 3.1.1.1 Réseau

La configuration des informations du réseau est définie dans le groupe « **networkInfos** ».

```
"networkInfos": [
  {
    "label": "Réseau d'eau",
    "name": "water",
    "webmap": "7c8e7f56259345f1953e379f387642c6",
    "webmap_mobile": "7201a1ba8b3b4c5782ec49983367d530",
    "networkService": "default:/1WATER_ORCL/1DevWater",
    "exploitationService": "default:/1WATER_ORCL/1OrclExploitation",
    "sectorizationService": "default:/1WATER_ORCL/1OrclSectorisation",
    "domainNetworkNames": [
      "RES_EAU"
    ]
  }
],
```

Figure 1 : Informations sur le réseau de distribution (Eau)

- **name** : Nom du réseau ;
- **label** : libellé qui s'affiche dans l'IHM ;
- **webmap** : identifiant de la carte ;
- **webmap\_mobile** : Si elle est remplie, cette clé désigne la carte à utiliser pour le mobile, si la clé est vide ou n'existe, pas la même carte est utilisée pour le web et le mobile
- **networkService** : URL d'accès aux services Utility Network ;
- **exploitationService** : URL d'accès aux services d'exploitation ;
- **sectorizationService** : URL d'accès aux services de sectorisation ;
- **domainNetworkNames** : liste des domaines du réseau.

### 3.1.1.2 Carte - Outil de localisation

La configuration de l'outil de localisation est définie dans le groupe « **localisation** » de « **map** ».

En plus de données issues de tiers il est possible de prédéfinir des recherches.

```
"localisation": {
  "sources": [
    {
      "name": "BAN",
      "featureServiceUrl": "default:/1WATER_ORCL/1OrclWater",
      "layerId": 150,
      "searchFields": [
        "subnetworkname"
      ],
      "displayField": "subnetworkname",
      "outFields": [
        "subnetworkname"
      ],
      "name": "Etage de pression par nom",
      "filter": {
        "where": "tiername=1"
      }
    }
  ],
  "minSuggestCharacters": 3,
  "maxSuggestions": 5
},
```

Figure 2 : Localisation (Eau)

```
"localisation": {
  "sources": [
    {
      "name": "BAN",
      "options": {
        "postcode": 56300
      }
    },
    {
      "featureServiceUrl": "default:/1WATER_DEV/1DevWater",
      "layerId": 150,
      "searchFields": [
        "subnetworkname"
      ],
      "displayField": "subnetworkname",
      "outFields": [
        "subnetworkname"
      ],
      "name": "Etage de pression par nom",
      "filter": {
        "where": "tiername=1"
      }
    }
  ],
  "minSuggestCharacters": 3,
  "maxSuggestions": 5
},
```

Figure 3 : Localisation – Restriction de la zone de recherche (Eau)

- **sources** : Listes des moteurs de recherche disponibles. Les valeurs de la liste peuvent être :

- « BAN » : Base d'adresse Nationale française. Ce type de moteur de recherche peut être configuré de deux manières :

- Soit en renseignant la chaîne de caractères « BAN » (valeur par défaut)
- Soit en renseignant un objet afin de restreindre le périmètre de recherche :
  - **name** : nom de la recherche = « BAN »
  - **options** : options permettant d'indiquer une commune.
  - **postcode** : code postal de la commune. Un seul code postal est possible
  - **citycode** : code INSEE de la commune. Un seul code INSEE est possible

Exemple :

```
{  
  "name": "BAN",  
  "options": {  
    "postcode": 56800  
  }  
}
```

- « OSM » : Base d'adresse d'OpenStreetMap. Ce type de moteur de recherche peut être configuré de deux manières :

- Soit en renseignant la chaîne de caractères « OSM » (valeur par défaut)
- Soit en renseignant un objet afin de restreindre le périmètre de recherche :
  - **name** : nom de la recherche = « OSM »
  - **options** : options permettant de renseigner la zone :
  - **viewcode** : cadrage correspondant à la zone de recherche, « x1, y1, x2, y2 ». Les valeurs sont à renseigner en WGS84 (x pour la longitude et y pour la latitude)

Exemple :

```
{  
  "name": "OSM",  
  "options": {  
    "viewcode": "-2.44846,47.86350,-2.30722,47.97302"  
  }  
}
```

- « Objet » décrivant les informations pour une recherche au niveau des objets d'une couche :

- **featureServiceUrl** : Url vers le service de la couche ;
- **layerId** : Identifiant de la couche (au niveau du service) ;
- **searchFields** : La liste des noms de champs de la couche à rechercher ;
- **displayField** : Nom du champ utilisé pour l'affichage des résultats. Par défaut, le « displayField » indiqué au niveau de la couche ;
- **outFields** : Spécifie les champs retournés avec les résultats de la recherche ;
- **name** : Nom de la recherche proposée à l'utilisateur ;
- **filter** : Filtre à appliquer lors de la recherche (where clause).

- **minSuggestCharacters** : Le nombre minimum de caractères à encoder avant que la recherche ne soit exécutée ;
- **maxSuggestions** : Le nombre maximum de valeurs proposées dans la liste.



### 3.1.1.3 Carte – Impression

La configuration de l'outil d'impression est définie dans le groupe « **print** » de « **map** ».

```
"map": {  
  "localisation": {  
    "print" : {  
      "printServiceUrl": "default:/Utilities/PrintingTools/GPServer/Export%20Web%20Map%20Task",  
      "title": null,  
      "author": null,  
      "copyright": null,  
      "legendEnabled": true,  
      "scaleEnabled": false,  
      "allowedFormats": ["pdf", "jpg", "png32", "gif"],  
      "allowedLayouts": ["a4-portrait", "a4-landscape", "a3-portrait", "a3-landscape"],  
      "dpi": null,  
      "northArrowEnabled": false  
    }  
  },  
}
```

*Figure 4 : Impression (Eau)*

- ▶ **printServiceUrl** : Url vers le service d'impression.
- ▶ **title** : libellé à afficher comme titre par défaut de l'impression. Cette valeur est modifiable par l'utilisateur.
- ▶ **author** : libellé à renseigner comme auteur par défaut de l'impression. Si aucune valeur n'est fournie, le nom de l'utilisateur qui a demandé l'impression est affiché et utilisé dans l'application. Cette valeur est modifiable par l'utilisateur.
- ▶ **copyright** : libellé du copyright par défaut à afficher dans l'impression. Cette valeur est modifiable par l'utilisateur.
- ▶ **legendEnabled** : indique si une légende doit être affichée dans le résultat de l'impression. Cette valeur est modifiable par l'utilisateur.
- ▶ **scaleEnabled** : indique si l'impression se base sur l'échelle courante de la carte (false) ou sur la valeur de l'échelle renseignée dans le champ « échelle » dans l'application (true). Cette valeur est modifiable par l'utilisateur.
- ▶ **allowedFormats** : liste des formats d'impression disponibles.
- ▶ **allowedLayouts** : liste des canevas d'impression disponibles.
- ▶ **dpi** : résolution de la carte imprimée (exprimée en dots per inch). Si aucune valeur n'est fournie, la valeur par défaut (96 dpi) est affichée et utilisée dans l'application. Cette valeur est modifiable par l'utilisateur.
- ▶ **northArrowEnabled** : indique si l'orientation du nord doit être affichée.

### 3.1.1.4 Identifiants des couches

La liste des couches utilisées par le module est définie dans le groupe « **layers** » :

```
"layers": {  
  "anomalyLayerId": 201,  
  "cutoffLayerId": 101,  
  "cutoffItemLayerId": 102,  
  "cutoffSubscriberLayerId": 100,  
  "cutoffNotificationLayerId": 103,  
  "leakSearchLayerId": 710,  
  "listenerLayerId": 712,  
  "listenerCoverageLayerId": 713,  
  "leakSearchPointsLayerId": 711,  
  "sensorTypeLayerId": 700,  
  "sensorAbacusLayerId": 701,  
  "sensorLayerId": 702,  
  "sensorSupportLayerId": 703,  
  "mesureLayerId": 714,  
  "scoringLayerId": 603,  
  "scoringPointsLayerId": 604,  
  "scoreLayerId": 605,  
  "currentScoreLayerId": 606,  
  "frameLayerId": 1001,  
  "consequenceLayerId": 1000,  
  "graffitiLayerId": 1002  
},
```

*Figure 5 : Couches d'exploitation (Eau)*

- ▶ **anomalyLayerId** : identifiant de la couche qui représente une anomalie dans les services 1Water. Ce sont les objets de cette couche qui apparaissent dans la carte lorsque le module des anomalies est actif ;
- ▶ **cutoffLayerId** : identifiant de la couche qui représente une coupure dans les services 1Water. Ce sont les objets de cette couche qui apparaissent dans la carte lorsque le module des coupures est actif ;
- ▶ **cutoffItemLayerId** : identifiant de la couche qui représente un élément de coupure dans les services 1Water ;
- ▶ **cutoffSubscriberLayerId** : identifiant de la couche des compteurs d'abonnés gérée dans le module des coupures ;
- ▶ **cutoffNotificationLayerId** : identifiant de la couche de suivi des envois de mail/sms gérée dans le module des coupures ;
- ▶ **leakSearchLayerId** : identifiant de la couche qui représente les recherches de fuite dans les services 1Water ;
- ▶ **listenerLayerId** : identifiant de la couche qui représente les écouteurs dans les services 1Water ;
- ▶ **listenerCoverageLayerId** : identifiant de la couche qui représente la couverture d'écoute dans les services 1Water ;
- ▶ **leakSearchPointsLayerId** : identifiant de la couche qui représente les points de recherche de fuite dans les services 1Water ;
- ▶ **sensorTypeLayerId** : identifiant de la couche qui représente les types capteurs dans les services 1Water ;
- ▶ **sensorAbacusLayerId** : table contenant la liste des abaques des capteurs ;
- ▶ **sensorLayerId** : identifiant de la couche qui représente les capteurs dans les services 1Water ;
- ▶ **sensorSupportLayerId** : identifiant de la couche qui représente les équipements pouvant supporter un capteur dans les services 1Water ;

- ▶ **mesureLayerId** : identifiant de la couche qui représente les mesures des capteurs dans les services 1Water ;
- ▶ **scoringLayerId** : identifiant de la couche qui représente un projet de notation dans les services 1Water. Ce sont les objets de cette couche qui apparaissent dans la carte lorsque le module des notations est actif ;
- ▶ **scoringPointsLayerId** : identifiant de la couche qui représente le centre géographique d'un projet de notation dans les services 1Water. Ce sont les objets de cette couche qui apparaissent dans la carte sous forme de pins lorsque le module des notations est actif ;
- ▶ **scoreLayerId** : identifiant de la couche qui représente une note calculée dans les services 1Water. Une note se décompose en notes et/ou critères ;
- ▶ **currentScoreLayerId** : identifiant de la couche qui représente une note calculée et validée dans les services 1Water. Une note se décompose en notes et/ou critères.
- ▶ **frameLayerId** : identifiant de la couche qui représente les cadrages ou les emplacements d'une zone de la carte. Ces cadrages ne sont enregistrés qu'en mobilité.
- ▶ **consequenceLayerId** : identifiant de la couche de gestion des différences observées au niveau du patrimoine entre les valeurs de la base de données et les valeurs constatées sur le terrain ou au travers des inspections télévisées.
- ▶ **graffitiLayerId** : Identifiant de la couche qui représente les graffitis ou annotations graphiques géolocalisés.

La liste des couches de sectorisation, utilisées par le module pour le calcul des notes, est définie dans le groupe « **sectorizationlayers** » :

```
"sectorizationlayers": {  
  "scoreGeoZone": 600,  
  "LevelScoreGeoZone": 601  
}
```

Figure 6 : Couches de sectorisation (Eau)

- ▶ **scoreGeoZone**: identifiant de la couche de zonage pour le calcul des notes ;
- ▶ **LevelScoreGeoZone** : identifiant de la couche de zonage par niveau pour le calcul des notes.

La liste des couches de dépose utilisées par le module est définie dans le groupe « **businessdatalayers** » :

```
"businessdatalayers": {  
  "wRemovalZone": 2100,  
  "wRemovalPipe": 2110,  
  "wRemovalJunction" : 2120,  
  "wRemovalDevice" : 2130,  
  "wRemovalAssembly" : 2140  
}
```

Figure 7 : Couches de dépose (Eau)

- ▶ **wRemovalZone**: identifiant de la couche de zonage des déposes ;
- ▶ **wRemovalPipe**: identifiant de la couche de linéaire déposé ;
- ▶ **wRemovalJunction**: identifiant de la couche de jonction déposée ;
- ▶ **wRemovalDevice**: identifiant de la couche d'appareil déposé ;
- ▶ **wRemovalAssembly**: identifiant de la couche d'assemblage déposé.

### 3.1.1.5 Rapports

La configuration des rapports est définie dans le groupe « **reports** » :

```
"reports": {  
  "cutoffNotificationPattern": {  
    "filepath": "RAPPORT_ARRET/NOTIFICATION_MAIL_STD_PATTERN"  
  },  
  "cutoffAnnouncement": {  
    "filepath": "RAPPORT_ARRET/AVIS_ARRET"  
  },  
  "cutoff": {  
    "filepath": "CUTOFF"  
  },  
  "deviceSynthesis": {  
    "filepath": "EQPT_EAU",  
    "maxScale": 2000  
  }  
}
```

*Figure 8 : Rapports (Eau)*

- **cutoffNotificationPattern** : Contenu du courriel et du sms envoyé lors de la coupure. Voir chapitre « **Notification** » dans le document « **1BiZ Server Guide Administrateur** » :
  - **filepath** : Nom du fichier contenant la définition des contenus pour la notification.
- **cutoffAnnouncement** : Génère un avis d'arrêt d'eau de la coupure :
  - **filepath** : Nom du fichier qui permet de générer le rapport précédé du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.
- **cutoff** : Affiche un rapport détaillé de la coupure :
  - **filepath** : Nom du fichier qui permet de générer le rapport précédé du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.
- **deviceSynthesis** : Génère une fiche de vie de l'équipement :
  - **filepath** : Nom du fichier qui permet de générer le rapport précédé du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.
  - **maxScale** : Echelle maximale de cadrage pour le plan de situation associé au rapport.

### 3.1.1.6 Service d'impression

La configuration du service d'impression est définie dans le groupe « **printService** » :

```
"printService": {  
  "url": "default:/Utilities/PrintingTools/GPServer/Export%20Web%20Map%20Task"  
},
```

*Figure 9 : Service d'impression (Eau)*

- **url** : Url vers le service d'impression ;

### 3.1.1.7 Tâches asynchrones

#### 3.1.1.7.1 Tâche asynchrone de gestion des suppressions (dépose)

Une tâche planifiée asynchrone pour le métier « Eau potable » permet, à la suite de la suppression et dans le cadre ou non d'une dépose d'un ou plusieurs éléments (tronçon, appareil, ...), de mettre automatiquement à jour l'état des anomalies qui sont liées au(x) élément(s) supprimé(s).

La configuration de cette tâche est définie dans le groupe « **parameters** » :

```
"parameters": {  
  "syncRemovalCron": "0 0 23 * * *",  
  "syncCsqCron": "0 0 21 * * *",
```

```
"maxUploadAttachmentSize": 5  
},
```

- **syncRemovalCron** : fréquence d'exécution de la tâche. La syntaxe de ce paramètre correspond au principe du cron UNIX/LINUX (seconde 0-59, minute 0-59, heure 0-23, jour du mois 1-31, mois 1-12, jour de la semaine 0 pour dimanche à 6 pour samedi).

**Exemples :**

- 0 0 23 \* \* \* : chaque jour à 23 heures 00 minutes 00 secondes ;
- 0 0 \* \* \* \* : chaque heure de chaque jour ;
- 0 0 8-10 \* \* \* : 8, 9 et 10 heures chaque jour à 8,9 et 10 heures ;
- 0 0 6,19 \* \* \* : chaque jour à 6 heures et à 19 heures ;
- 0 0/30 8-10 \* \* \* : chaque jour à 8:00, 8:30, 9:00, 9:30, 10:00 et 10:30 ;
- 0 0 9-17 \* \* MON-FRI : toutes les heures entre 9 heures et 17 heures du lundi au vendredi ;
- 0 0 0 25 12 ? : chaque 25 décembre à minuit ;
- 0 0 0 L \* \* : dernier jour du mois à minuit ;
- 0 0 0 \* \* THUL : dernier jeudi du mois à minuit ;
- 0 0 0 ? \* 5#2 : deuxième vendredi du mois à minuit.

! Si la clé **syncRemovalCron** n'est pas présente la tâche asynchrone n'est pas déclenchée.

### 3.1.1.7.2 Tâche asynchrone de la remontée des conséquences

Certains modules permettent de corriger la valeur d'attributs du patrimoine de l'eau. Pour des raisons de cohérence et d'efficacité, ces corrections ne sont pas réalisées directement mais par l'intermédiaire d'une tâche asynchrone.

La configuration de cette tâche est définie dans le groupe « **parameters** » :

```
"parameters": {  
  "syncRemovalCron": "0 0 23 * * *",  
  "syncCsqCron": "0 0 21 * * *",  
  "maxUploadAttachmentSize": 5  
},
```

- **syncCsqCron** : fréquence d'exécution de la tâche. La syntaxe de ce paramètre correspond au principe du cron UNIX/LINUX (seconde 0-59, minute 0-59, heure 0-23, jour du mois 1-31, mois 1-12, jour de la semaine 0 pour dimanche à 6 pour samedi).

**Exemples :**

- 0 0 23 \* \* \* : chaque jour à 23 heures 00 minutes 00 secondes ;
- 0 0 \* \* \* \* : chaque heure de chaque jour ;
- 0 0 8-10 \* \* \* : 8, 9 et 10 heures chaque jour à 8,9 et 10 heures ;
- 0 0 6,19 \* \* \* : chaque jour à 6 heures et à 19 heures ;
- 0 0/30 8-10 \* \* \* : chaque jour à 8:00, 8:30, 9:00, 9:30, 10:00 et 10:30 ;
- 0 0 9-17 \* \* MON-FRI : toutes les heures entre 9 heures et 17 heures du lundi au vendredi ;
- 0 0 0 25 12 ? : chaque 25 décembre à minuit ;
- 0 0 0 L \* \* : dernier jour du mois à minuit ;
- 0 0 0 \* \* THUL : dernier jeudi du mois à minuit ;

- 0 0 0 ? \* 5#2 : deuxième vendredi du mois à minuit.

! Si la clé **syncCsqCron** n'est pas présente la tâche asynchrone n'est pas déclenchée.

### 3.1.1.8 Pièces jointes

La configuration des pièces jointes est définie dans le groupe « **parameters** » :

```
"parameters": {  
  "syncRemovalCron": "0 0 23 * * *",  
  "syncCsqCron": "0 0 21 * * *",  
  "maxUploadAttachmentSize": 5  
},
```

**maxUploadAttachmentSize** : Taille maximum autorisée d'un fichier à télécharger en tant que pièce jointe. Cette valeur est à renseigner en Mégabytes.

! Les pièces jointes sont copiées dans la base et impacte donc sa volumétrie.

### 3.1.1.9 Modèle numérique de terrain

Le modèle numérique de terrain est exploité par le webservice de calcul d'altitude et par le profil en long.

La configuration du modèle numérique de terrain est définie dans le groupe « **mnt** » :

```
"mnt": {  
  "accessor": "gdal",  
  "source": "gdal001/morbihan.vrt"  
}
```

*Figure 10 : Modèle numérique de terrain (Eau)*

- **accessor** : indique le logiciel d'accès aux données du MNT (uniquement gdal dans cette version) ;
- **source** : indique le chemin du fichier « .vrt ». Le chemin est relatif, il s'appuie sur le répertoire « <MonServeur>\config\permanent\mnt ».

! Les données MNT doivent être dans la même référence spatiale que les données ESRI.

## 3.1.2 Anomalie

Ce module permet de créer ou supprimer une anomalie, de renseigner les informations de cette dernière ainsi que les informations relatives à l'intervention qui a été effectuée pour la réparer.

Le module « Anomalies » est associé à des droits fonctionnels (consultation, édition et gestion). Il est nécessaire de posséder au moins le droit de consultation pour accéder à ce module.

L'utilisateur peut également être associé à une restriction géographique pour contrôler ses actions d'édition et de gestion.

### 3.1.2.1 Configuration

La configuration du module « Anomalie » est définie dans le fichier « **\config\onewater\water\anomaly.json** » :

```
{  
  "maxItemsInList": 50,  
  "canManageCutOff": true,  
  "networkEntities": [  
    "filters": [  
    ]  
  ]  
}
```

Figure 11 : Anomalie (Eau)

- **maxItemsInList** : nombre maximum d'anomalies qu'il est possible d'afficher dans la carte et dans la liste des anomalies. Si dans l'étendue courante de la carte il y a plus d'anomalies que la valeur de « maxItemsInList » :
  - L'affichage des anomalies dans la carte bascule en mode « cluster » ;
  - La liste contient un message qui indique notamment le nombre d'anomalies présentes (en lieu et place de la liste des anomalies présentes dans la carte).
- ! Plus la valeur de ce paramètre est élevée, plus la fluidité d'affichage de la carte risque d'être dégradée.
- **canManageCutOff** : autorisation de la gestion de la coupure ;
- **networkEntities** : Informations sur les types d'équipements qui peuvent faire l'objet d'une anomalie.  
Pour chaque type :
  - **source** : type d'usage de l'équipement au niveau du réseau utility network.  
Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction ».
  - **assetGroup** : code de l'assetGroup correspondant à l'équipement au sein de la couche concernée.
  - **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné.
  - **typeRestrictions** : Liste des codes correspondant aux types d'anomalie à proposer pour ce type d'équipement (filtre qui permet de faire une dépendance entre le type d'équipement choisi et les types d'anomalies proposés pour ce type d'équipement).

```
"networkEntities": [  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 100,  
    "domainNetworkName": "RES_EAU",  
    "typeRestrictions": [  
      1,  
      2  
    ]  
  },  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 101,  
    "domainNetworkName": "RES_EAU",  
    "typeRestrictions": [  
      2  
    ]  
  },  
  {  
    "source": "esriUNFCUTJunction",  
    "assetGroup": 120,  
    "domainNetworkName": "RES_EAU",  
    "typeRestrictions": [  
      2  
    ]  
  },  
  {  
    "source": "esriUNFCUTJunction".  
  }  
]
```

Figure 12 : Patrimoine pour anomalie (Eau)



! Si cette liste est mal configurée (vide ou aucune valeur correspondant au modèle de données), le module est uniquement accessible en consultation.

► **filters** : liste des filtres de sélections personnalisés. Un filtre de sélection personnalisé permet d'affiner la portée de la sélection qui est faite dans la carte. Au lieu de sélectionner tous les équipements « sélectionnables », on ne va sélectionner que ceux qui répondent à la condition liée au filtre personnalisé actif.  
Par exemple :

- Uniquement les équipements ponctuels
- Uniquement les vannes
- Uniquement les tronçons de type « distribution »
- Uniquement les équipements qui sont visitables
- Uniquement les tronçons qui sont en PVC, en acier ou en fonte et qui ont une longueur supérieure à 40m
- ...

Des exemples de filtres personnalisés sont disponibles en [annexe](#).

Pour chaque filtre personnalisé :

- **name** : nom du filtre (obligatoire).
- **description** : description du filtre. Optionnel mais recommandé. Permet à l'utilisateur de savoir quelle est la portée du filtre et la(les) condition(s) appliquée(s)
- **condition** : condition à appliquer sur les équipements du réseaux lors de la sélection
  - **networkEntities** : informations sur les types d'équipements qui peuvent être sélectionnés. Doit être un sous-ensemble du *networkEntities* déclaré au niveau supérieur. Optionnel.  
Pour chaque type :
    - **source** : type d'usage de l'équipement au niveau du réseau utility network.  
Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction » ;
    - **assetGroup** : code de l'assetGroup correspondant à l'équipement au sein de la couche concernée ;
    - **assetType** : code de l'assetType correspondant au type d'équipement de l'assetGrout concerné. Exemple : tronçon (= assetGroup) de type refoulement, décharge, simple, etc... (=assetType) ;
    - **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné ;
      - **whereClause** : condition qui s'applique soit sur les types d'équipements renseignés dans *networkEntities* (si *networkEntities* est renseigné) soit sur tous les types d'équipements sélectionnables pour le module (cfr. *networkEntities* du niveau supérieur).

### 3.1.3 Coupure d'eau

Ce module permet de simuler une coupure d'eau, de la créer ensuite, et d'historiser les coupures d'eau réalisées pour intervenir sur le réseau.

Chaque coupure permet d'identifier les éléments à isoler (conduites), les éléments coupants (vannes) et les impacts (abonnés et équipements incendie) pour ensuite la planifier, alerter (avis d'arrêt d'eau, envoi de mail et de sms) ou s'interfacer avec votre système d'information.

La configuration de la propagation s'appuie sur les catégories et les attributs de réseau défini au niveau de l'Utility Network.

La configuration du module « Coupure d'eau » est définie dans le fichier « `\config\onewater\water\cutoff.json` ».

### 3.1.3.1 Configuration de la trace

Une simulation de coupure est exécutée par le « Utility Network », qui effectue une propagation à partir d'un élément du réseau choisi par l'utilisateur (Trace).

La partie « **traceConfiguration** » permet de définir la propagation :

```
"traceConfiguration": {  
  "traceIsolationNames": [  
    "1W_CUTOFF_ISOLATION"  
  ],  
  "traceConnectedNames": [  
    "1W_CUTOFF_CONNECTED"  
  ],  
  "traceAbortConditions": {  
    "maxBarriers": 500,  
    "maxEdges": 500  
  }  
},
```

Figure 13 : Coupure - Trace

- **traceIsolationNames** : Nom de la trace utilisée pour rechercher les éléments coupants ;
- **traceConnectedNames** : Nom de la trace utilisée pour rechercher les éléments coupés ;
- **traceAbortConditions** : Permet de spécifier des limites à la propagation, au-delà desquelles une propagation est considérée comme erronée à la suite de configurations dans « Utility Network » :
  - **maxBarriers** : Limite maximale du nombre de barrières qu'une propagation peut produire ;
  - **maxEdges** : Limite maximale du nombre de tronçons qu'une propagation peut produire.

La configuration de la propagation s'appuie sur les catégories et les attributs du réseau défini au niveau de l'Utility Network.

### 3.1.3.2 Filtres de la trace

La coupure s'appuie sur un ensemble de filtres, basés sur la notion de catégories du réseau de distribution, pour regrouper les différents éléments :

```
"startingPointFilters": {  
  "categoryNames": [  
    "HANDLER",  
    "CUTOFF_LINEAR"  
  ],  
  "cutRole": 1  
},  
"edgeFilters": {  
  "categoryNames": [  
    "CUTOFF_LINEAR"  
  ]  
},  
"servicePointFilters": {  
  "categoryNames": [  
    "CLIENTPOINT",  
    "SERVICEPOINT"  
  ]  
},  
"barrierFilters": {  
  "cutRole": 2  
},  
"extensionFilters": {  
  "cutRole": 3  
},  
"geometryFilters": {  
  "categoryNames": [  
    "CUTOFF_LINEAR"  
  ]  
},
```

Figure 14 : Coupure - Filtres

- **startingPointFilter** : Permet de définir les points de départ possibles pour la coupure :

- **categoryNames** : liste des catégories du réseau de distribution auxquelles sont associés les éléments sélectionnables comme point de départ de la trace ;
  - **cutRole** : Valeur fixée à 1 (Ne pas modifier).
  - ▶ **edgeFilters** : Permet de définir les éléments qui sont à prendre en compte dans le linéaire impacté par la coupure dans le rapport :
    - **categoryNames** : liste des catégories du réseau de distribution auxquelles sont associés les éléments linéaires impactés par la coupure.
  - ▶ **servicePointFilters** : Permet de définir les éléments de service impactés par la coupure :
    - **categoryNames** : liste des catégories du réseau de distribution auxquelles sont associés les éléments de service impactés par la coupure.
  - ▶ **barrierFilters** : Permet de définir les éléments isolants de la coupure :
    - **cutRole** : Valeur fixée à 2 (Ne pas modifier).
  - ▶ **extensionFilters** : Permet de définir les éléments d'extension de la coupure. Les éléments d'extension sont issus des éléments isolants :
    - **cutRole** : Valeur fixée à 3 (Ne pas modifier).
  - ▶ **geometryFilters** : Permet de définir les éléments qui sont à prendre en compte pour créer la géométrie de la coupure :
    - **categoryNames** : liste des catégories du réseau de distribution auxquelles sont associés les éléments linéaires pour créer la géométrie de la coupure.
- ! Comme indiqué dans la description ci-dessus, les balises « cutRole » ne doivent pas être modifiées.

### 3.1.3.3 Éléments impactés

Le résultat de la coupure permet de mettre en évidence les éléments qui seront impactés par celle-ci. Ces éléments sont historisés afin de pouvoir, par la suite, les exploiter même si le réseau a été modifié.

La partie « **impactedElementFilters** » permet de définir ces éléments et leur répartition :

```
"impactedElementFilters": {
  "sensibleElements": {
    "clientPoint": {
      "mainCategoryName": "CLIENTPOINT",
      "cutRole": 4
    },
    "fireEquipment": {
      "mainCategoryName": "FIRE EQUIPMENT",
      "auxiliaryCategoryName": "SERVICEPOINT",
      "cutRole": 5
    },
    "others": [
      {
        "mainCategoryName": "SERVICEPOINT"
      }
    ]
  },
  "notSensibleElements": {
    "others": [
      {
        "mainCategoryName": "CUTOFF_LINEAR"
      }
    ]
  }
}
```

Figure 15 : Coupure - Éléments impactés

- **sensibleElements** : Permet de définir les éléments impactés et considérés comme sensibles, qui sont retournés par la trace de coupure. Ces éléments sont mis en évidence et présentés dans les listes et dans le rapport :
  - **clientPoint** : Permet de définir les points de livraison client.
    - **mainCategoryName** : nom de la catégorie du réseau de distribution à laquelle sont associés les points de livraison. Dans cette version, la notion de sensible des points de livraison n'est pas configurable. Un point de livraison sensible appartient à la catégorie donnée et est associé à au moins un abonné sensible (table « E-Abonné » avec le champ « sensible » égal à 1).
    - **cutRole** : Valeur fixée à 4 (Ne pas modifier).
  - **fireEquipment** : Permet de définir les équipements incendie.
    - **mainCategoryName** : nom de la catégorie principale du réseau de distribution à laquelle sont associés les équipements incendie.
    - **auxiliaryCategoryName** : nom de la catégorie secondaire du réseau de distribution à laquelle sont associés les équipements incendie.
    - **cutRole** : Valeur fixée à 5 (Ne pas modifier).
  - **others** : Permet de définir d'autres éléments :
    - **mainCategoryName** : nom de la catégorie du réseau de distribution à laquelle sont associés les autres équipements.
- **notSensibleElements** : Permet de définir les éléments impactés, et considérés comme non sensibles, qui sont retournés par la trace de coupure. Ces éléments sont présentés dans le rapport.
  - **others** : Permet de définir les autres éléments considérés comme non sensible.

- **mainCategoryName** : nom d'une catégorie du réseau de distribution à laquelle sont associés les autres équipements.

! Comme indiqué dans la description ci-dessus, les balises « cutRole » ne doivent pas être modifiées.

### 3.1.3.4 Notifications

Pour les coupures à l'état « Planifiée », un service web, décrit dans le chapitre « Notification » du document « **1BiZ Server Guide Administrateur** », permet d'envoyer un mail et ou un SMS aux abonnés concernés.

Afin de rendre le plus paramétrable possible le contenu des mails et SMS d'avis de coupure, les éléments sont produits dans un rapport Jasper.

Ce rapport comprend plusieurs paramètres :

- ▶ **PARA\_START\_DATE** : Date complète de début de la coupure ;
- ▶ **PARA\_END\_DATE** : Date complète de fin de coupure ;
- ▶ **PARA\_ID\_PDL** : Identifiant unique du point de livraison ;
- ▶ **PARA\_REF\_CLIENT** : Référence du client ;
- ▶ **PARA\_CUTOFF\_ID** : Identifiant unique de la coupure.

Les éléments produits par le rapport sont :

- ▶ le sujet du mail ;
- ▶ le corps du mail ;
- ▶ le contenu du SMS.

Les trois éléments sont détectés dans le texte du rapport produit de la façon suivante :

- ▶ le sujet du mail est le texte compris entre les balises **\$\$SUBJECT\$\$** et **\$\$** ;
- ▶ le corps du mail est le texte compris entre les balises **\$\$BODY\$\$** et **\$\$** ;
- ▶ le contenu du SMS est le texte compris entre les balises **\$\$SMS\$\$** et **\$\$**.

Le rapport par défaut se nomme « **NOTIFICATION\_MAIL\_STD\_PATTERN.jrxml** » et est situé dans le répertoire « **config/reports/onewater/water/cutoff/RAPPORT\_ARRET** ». Ce rapport est modifiable afin de pouvoir s'adapter aux besoins de l'utilisateur.

## 3.1.4 Export de données

### 3.1.4.1 Configuration d'export de données

La configuration du module « Export de données » est définie dans le fichier « **\config\onewater\water\export.json** » au niveau de « **services** ».

```
{
  "services": [
    {
      "name": "EPANET",
      "url": "default:/{{{water.exportServices.EPANETUrl}}}"
    },
    {
      "name": "STARDT",
      "url": "default:/{{{water.exportServices.STARDTUrl}}}"
    }
  ]
}
```

*Figure 16 : Export (Eau)*

Chaque type d'export est défini par :

- **nom** : Nom associé au format d'export ;
- **url** : Url à appeler pour réaliser le traitement de conversion vers le format attendu.

## 3.1.5 Recherche de fuite

### 3.1.5.1 Configuration de la recherche de fuite

La configuration du module « Recherche de fuite » est définie dans le fichier « \config\onewater\water\leaksearch.json » :

```
"buffer": 5.0,
"maxItemsInList": 50,
"traceConfiguration": {
  "url": "default:/1WATER_ORCL/ComputeCover/GPServer/ListenerCover/execute"
}
```

*Figure 17 : Recherche de fuites*

- **Buffer** : valeur de la tolérance pour l'accroche par saisie lors de la pose d'un capteur ;
- **maxItemsInList** : nombre maximum de « Pré localisation » avant la bascule en mode cluster ;
- **traceConfiguration.url** : url de la fonction de trace associée à la pose d'un capteur d'écoute.

## 3.1.6 Profil en long

Ce module permet de réaliser des profils en long sur le réseau d'eau.

La configuration du module « Profil en long » est définie dans le fichier « \config\onewater\water\longprofile.json ».

Pour plus de détails, consulter le chapitre « [Profil en long](#) » dans la partie « Annexes ».



## 3.1.7 Notation

### 3.1.7.1 Configuration du module

La configuration du module « Notation » est définie dans le fichier « \config\onewater\water\scoring.json » :

```
"buffer": 5.0,
"networkEntities": [
  {
    "source": "esriUNFCUTLine",
    "assetGroup": 100,
    "domainNetworkName": "RES_EAU"
  }
],
"filters": [
  {
    "name": "Distribution",
    "description": "Uniquement tronçons de distribution",
    "condition": {
      "networkEntities": [
        {
          "source": "esriUNFCUTLine",
          "assetGroup": 100,
          "assetType": 2,
          "domainNetworkName": "RES_EAU"
        }
      ]
    }
  },
  {
    "name": "Production",
    "description": "Uniquement tronçons de production",
    "condition": {
      "networkEntities": [
        {
          "source": "esriUNFCUTLine",
          "assetGroup": 100,
          "assetType": 1,
          "domainNetworkName": "RES_EAU"
        }
      ]
    }
  }
],
"geographicSearchSources": [
  {
    "featureServiceUrl": "default:/1WATER_DEV/1DevSectorisation",
    "layerId": 1
  }
]
```

Figure 18 : Notation (Eau)

► **buffer** : taille en mètre de la zone tampon autour des ouvrages concernés par la notation.

Pour définir un type d'élément (au niveau de la section « **networkEntities** ») :

- **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné ;
- **source** : type d'usage de l'équipement au niveau du réseau utility network. Valeurs possibles : « esriUNFCUTLine » ;
- **assetGroup** : code de l'assetGroup correspondant à l'équipement au sein de la couche concernée.

! Si cette liste est mal configurée (vide ou aucune valeur correspondant au modèle de données), le module est uniquement accessible en consultation.

Pour définir les objets utilisés comme source de la sélection géographique (au niveau de la section « **geographicSearchSources** ») :

- **featureServiceUrl** : Url vers le service de la couche ;
- **layerId** : Identifiant de la couche (au niveau du service) ;

La section « **filters** » permet de définir la liste des filtres de sélection personnalisés. Un filtre de sélection personnalisé permet d'affiner la portée de la sélection qui est faite dans la carte. Au lieu de sélectionner tous les équipements « sélectionnables », on ne va sélectionner que ceux qui répondent à la condition liée au filtre personnalisé actif. Par exemple :

- ▶ Uniquement les équipements linéaires
- ▶ Uniquement les équipements qui sont dans le domaine « eau usée »
- ▶ Uniquement les tronçons qui sont en PVC, en acier ou en fonte
- ▶ Etc...

Des exemples de filtres personnalisés sont disponibles en [annexe](#).

Pour chaque filtre personnalisé :

- ▶ **name** : nom du filtre (obligatoire).
- ▶ **description** : description du filtre. Optionnel mais recommandé. Permet à l'utilisateur de savoir quelle est la portée du filtre et la(les) condition(s) appliquée(s)
- ▶ **condition** : condition à appliquer sur les équipements du réseaux lors de la sélection
  - **networkEntities** : informations sur les types d'équipements qui peuvent être sélectionnés. Doit être un sous-ensemble du *networkEntities* déclaré au niveau supérieur. Optionnel. Pour chaque type :
    - **source** : type d'usage de l'équipement au niveau du réseau utility network. Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction » ;
    - **assetGroup** : code de l'assetGroup correspondant à l'équipement au sein de la couche concernée ;
    - **assetType** : code de l'assetType correspondant au type d'équipement de l'assetGroup concerné. Exemple : tronçon (= assetGroup) de type refoulement, décharge, simple, etc... (=assetType) ;
    - **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné ;
  - **whereClause** : condition qui s'applique soit sur les types d'équipements renseignés dans *networkEntities* (si *networkEntities* est renseigné) soit sur tous les types d'équipements sélectionnables pour le module (cfr. *networkEntities* du niveau supérieur).

### 3.1.7.2 Configuration des notes

La configuration des notes pour l'application est définie dans le fichier « `\config\onewater\water\score_definitions.json` » :

```
{  
  "name": "CASSE",  
  "description": "Note de casse",  
  "size": 4,  
  "productName": "onewater",  
  "moduleName": "water",  
  "formula": "STANDARD",  
  "criteria": [  
  ]  
}
```

*Figure 19 : Définition des notes (Eau)*

Pour la description de ce fichier de configuration, vous référer au document « **1Water Méthode de notation** ».

### 3.1.8 Mise à jour

Le module « Mise à jour » permet de saisir, de mettre à jour et de supprimer les objets du patrimoine dans le cadre d'une session de travail privée et dédiée à l'utilisateur. Cette session de travail est ensuite validée et publiée pour être partagée avec les autres utilisateurs.

La configuration, qui permet de définir des prototypes d'enchaînement par domaine du réseau de distribution, est décrite dans le fichier « **\config\onewater\water\update.json** ».

Exemple de prototype d'un point de livraison :

```

"RES_EAU": {
  "150": {
    "154": {
      "ALL": [
        {
          "source": "esriUNFCUTDevice",
          "name": "Création d'un point de livraison - Individuel",
          "steps": [
            {
              "name": "info",
              "label": "Informations",
              "fields": [
                {
                  "name": "ident",
                  "type": "input",
                  "label": "Numéro",
                  "defaultValue": "FDL_E-XXX",
                  "isRequired": true
                }
              ]
            },
            {
              "name": "draw",
              "label": "Dessiner",
              "message": "Dessiner un arc",
              "infos": [
                {
                  "label": "1er point:",
                  "value": "Position du piquage"
                },
                {
                  "label": "2e point:",
                  "value": "Position de la vanne de branchement"
                },
                {
                  "label": "Dernier point:",
                  "value": "Position du point de livraison"
                }
              ]
            }
          ]
        }
      ],
      "features": [
        {
          "source": "esriUNFCUTJunction",
          "assetGroup": 124,
          "assetType": 201,
          "connected": true
        },
        {
          "source": "esriUNFCUTLine",
          "assetGroup": 101,
          "assetType": 51
        },
        {
          "source": "esriUNFCUTDevice",
          "assetGroup": 163,
          "assetType": 654
        },
        {
          "source": "esriUNFCUTLine",
          "assetGroup": 101,
          "assetType": 51
        },
        {
          "source": "esriUNFCUTDevice",
          "assetGroup": 154,
          "assetType": 201,
          "fieldAttributes": "info"
        }
      ]
    }
  }
}

```

Figure 20 : Prototype « Point de livraison »

**RES\_EAU** : Identifiant du domaine du réseau de distribution concerné par la liste des prototypes, avec pour chaque prototype :

- **#ID\_GROUP#**: Identifiant du groupe permettant de positionner la liste des prototypes dans l'arborescence de l'application ;
- **#ID\_TYPE#** ou **"ALL"**: Identifiant du type permettant de positionner la liste de prototypes dans l'arborescence de l'application ou ALL qui le place au niveau chaque type;
  - **source** : Identifiant de la source permettant de positionner le prototype dans l'arborescence de l'application ;
  - **name** : Titre du panneau lors de l'ouverture du prototype ;
  - **steps** : Liste des étapes du prototype (renseignement des attributs, saisie graphique des points) :

- **name** : Nom de l'étape qui doit être unique ;
- **label** : Libellé de l'étape qui apparaît au niveau de la progression du prototype ;
- **message** : Message d'information affiché à l'utilisateur pour l'orienter dans l'action à effectuer (facultatif) ;
- **fields** (renseignement des attributs) : Liste des champs à renseigner, pour le groupe associé au prototype courant, avant la création :
  - **name** : Nom du champ qui doit être unique et correspondre à un nom d'attribut de l'assetGroup ;
  - **type** : Type du champ :
    - **assetTypes** : Liste déroulante des types disponibles pour le groupe du prototype ;
    - **input** : Champ texte éditable.
  - **label** : Libellé indiqué au niveau du champ ;
  - **defaultValue** : Valeur par défaut du champ ;
  - **isRequired** : Indique si le champ est obligatoire. Si le champ est obligatoire, l'étape ne pourra être validée sans renseigner le champ.
- **infos** (saisie graphique de point) : Informations sur la marche à suivre pour l'étape courante :
  - **label** : Etape de saisie ;
  - **value** : Message associé à l'étape de saisie.
  - **features** : Liste des objets à créer. Chaque saisie ponctuelle, définie dans « **infos** » des étapes (**steps**), correspond à la création d'un objet ponctuel déclaré dans ce bloc et en respectant l'ordre. Les éléments linéaires sont obligatoirement déclarés entre deux saisies ponctuelles :
- **source** : Identifiant de la source de l'objet à créer ;
- **assetGroup** : Identifiant du groupe de l'objet à créer ;
- **assetType** : Identifiant du type de l'objet à créer ;
- **attributes** : Attributs à renseigner par défaut de l'objet :
  - **<clé>** : Nom de l'attribut ;
  - **<valeur>** : Valeur de l'attribut.
- **fieldAttributes** : Nom (**name**) des étapes (**steps**) défini précédemment pour récupérer les couples « attribut / valeur » renseignées par l'utilisateur au niveau de « **fields** ».

Exemple de définition de prototypes différents par #ID\_TYPE# d'un même #ID\_GROUP# :

```
"RES_EAU": {  
  "150": {  
    "4": [  
      {  
        "source": "esriUNFCUTDevice",  
        "name": "Création d'un appareil de protection - Purge d'extrémité",  
        "steps": [  
          "features": [  
            ],  
          ],  
        },  
      ],  
    ],  
    "2": [  
      {  
        "source": "esriUNFCUTDevice",  
        "name": "Création d'un appareil de protection - Ventouse",  
        "steps": [  
          "features": [  
            ],  
          ],  
        },  
      ],  
    ],  
  },  
}
```

Figure 20: Prototypes par type d'un même groupe (Eau)

### 3.1.9 Cadastre

Des fonctionnalités, liées au cadastre, sont proposées sous la forme de widget et permettent :

- ▶ de se localiser par référence cadastrale ;
- ▶ de se localiser par lieu-dit ;
- ▶ de consulter les informations d'une parcelle cadastrale.

Pour bénéficier de ces fonctionnalités, il est nécessaire d'avoir un accès à la base de données cadastre configuré dans le fichier « \config\cadastre.properties » :

```
cadastre.database.type=postgresql  
cadastre.database.geometry.mode=pg_geometry  
cadastre.database.geometry.srid=2154  
cadastre.database.driver=org.postgresql.Driver  
cadastre.database.url=jdbc:postgresql://lbz-d-1wags11.1spatial.fr:5432/dev_water  
cadastre.database.user=cadastre  
cadastre.database.password=cadastre  
cadastre.database.initialSize=0  
cadastre.database.maxTotal=8  
cadastre.database.validationQuery=select version();  
cadastre.database.schema.cadastre=cadastre  
cadastre.database.schema.plu=cadastre  
urba.directory=E:\\DATA\\Cadastre\\urba
```

Figure 21 : Configuration du cadastre sous PostgreSQL (Eau)

! Le fichier de configuration peut prendre deux formes selon que la base de données « Cadastre » soit sous Oracle ou PostgreSQL.

La signification de chaque clé est la suivante :

- ▶ **cadastre.database.type** : indique le type de base de données cadastre utilisé. Les valeurs possibles sont :
  - oracle
  - postgresql
- ▶ **cadastre.database.geometry.mode** : mode de géométrie pour la base de données. Les valeurs possibles sont :
  - **st\_geometry** : géométrie d'ESRI ;
  - **pg\_geometry** : si « **cadastre.database.type** » est égal à « **postgresql** » ;
  - **sdo\_geometry** : si « **cadastre.database.type** » est égal à « **oracle** ».

- ▶ **cadastre.database.geometry.srid** : système de référence des géométries représentées dans la base de données.
- ▶ **cadastre.database.driver** : classe du driver de base de données. Les valeurs possibles sont :
  - **org.postgresql.Driver** : si « **cadastre.database.type** » est égal à « **postgresql** » ;
  - **oracle.jdbc.OracleDriver** : si « **cadastre.database.type** » est égal à « **oracle** ».
- ▶ **cadastre.database.url** : url de connexion à la base de données ; Selon le type de base de données la chaîne de connexion peut prendre différente forme
  - si « **cadastre.database.type** » est égal à « **postgresql** », elle est du type **jdbc:postgresql://<host>[<port>]/<nom de la base de données>** avec :
    - **<host>** : nom de la machine sur laquelle est installée la base de données ;
    - **<port>** : numéro du port. Cette valeur est optionnelle (par défaut 5432).
  - si « **cadastre.database.type** » est égal à « **oracle** », elle est du type **jdbc:oracle:thin:@//<host>[:<port>]/<service\_name>** avec :
    - **<host>** : nom de la machine sur laquelle est installée la base de données ;
    - **<port>** : numéro du port.
- ▶ **cadastre.database.user** : utilisateur de connexion à la base de données.
- ▶ **cadastre.database.password** : mot de passe de l'utilisateur de connexion. Le mot de passe de l'utilisateur de connexion peut être crypté.  
Le fichier « **Encrypt.bat** », présent au niveau du répertoire racine du serveur 1BiZ, permet d'encrypter un mot de passe.

```
C:\1BiZServer>Encrypt.bat abcd
cryptage...
abcd: <xMIIRx5J19InuoTO+0Pr2zx8N2YE7L2QPz1T+k03+Vo=>
```

- ! Copier le contenu entre <> au niveau de la clé dans le fichier de configuration et le faire précéder d'un ~ (tilde).

Exemple : **cadastre.database.password**=~xMIIRx5J19InuoTO+0Pr2zx8N2YE7L2QPz1T+k03+Vo=

- ▶ **cadastre.database.initialSize** : taille initiale du pool de connexion pour les connexions à la base de données.
  - ▶ **cadastre.database.maxTotal** : taille maximale du pool de connexion pour les connexions à la base de données.
  - ▶ **cadastre.database.validationQuery** : requête utilisée pour valider la connexion à la base de données. En général, on utilise :
    - **select version();** : si « **cadastre.database.type** » est égal à « **postgresql** » ;
    - **select 1 from dual;** : si « **cadastre.database.type** » est égal à « **oracle** ».
  - ▶ **cadastre.database.schema.cadastre** : nom du schéma associé au cadastre dans la base de données.
  - ▶ **cadastre.database.schema.plu** : nom du schéma contenant le plan d'occupation des sols (PLU).
  - ▶ **urba.directory** : chemin vers le répertoire contenant les documents d'urbanisme.
- ! Les clés, dans le fichier de configuration livré, ne sont pas renseignées par défaut. Elles doivent toutes être configurées pour accéder aux fonctionnalités du cadastre.

### 3.1.10 Gestion des conséquences

La fonctionnalité « Gestion des conséquences » n'est pas à proprement dit un module, mais une fonctionnalité « transverse ». Une conséquence correspond à une modification d'un attribut sur un élément du patrimoine.

Les conséquences proviennent d'informations remontées par la mobilité (ex. : lors de la réparation d'une anomalie).

La configuration de la fonctionnalité « Gestion des conséquences » est définie dans le fichier « **\config\onewater\water\consequences.json** » : par défaut, ce fichier contient une liste vide ( {} ).

Le format est le suivant :

```
{
  "networkEntities": [
    {
      "domainNetworkName": "<Nom du domaine de réseau>",
      "source": "<Nom de la source UN>",
      "assetGroup": "<Code de l'AssetGroup>",
      "attributes": [
        {
          "name": "<Nom de l'attribut>",
          "replaceMode": "<Mode de remplacement de la valeur de l'attribut>"
        }
      ]
    }
  ]
}
```

Les paramètres sont :

- **networkEntities** : la liste des types d'équipements qui peuvent faire l'objet d'une conséquence ;
  - **domainNetworkName** : Texte, obligatoire, pour le domaine de réseau;
  - **source** : Texte, obligatoire, pour la source;
  - **assetGroup** : Entier, obligatoire, pour le code de l'AssetGroup ;
  - **attributes** : la liste des attributs, pouvant faire l'objet d'une conséquence, pour l'AssetGroup ;
    - **name** : Texte (max 50 caractères), obligatoire, pour le nom de l'attribut ;
    - **replaceMode** : Entier, facultatif, pour le mode de remplacement lors de la gestion des conséquences sur le serveur :
      - 0 : la conséquence est systématiquement appliquée (valeur par défaut) ;
      - 1 : la conséquence est appliquée seulement si l'attribut n'est pas renseigné. Si l'attribut est renseigné et que sa valeur est différente de la valeur de la conséquence, une validation manuelle est nécessaire ;
      - 2 : la conséquence nécessite une validation manuelle sauf si la valeur de l'attribut est égale à celle de la conséquence.

Exemple de fichier de configuration de gestion des conséquences :

```
{
  "modifiableAssets": [
    {
      "domainNetworkName": "RES_EAU",
```



```
"source": "esriUNFCUTJunction",
"assetGroup": 124,
"modifiableAttributes": [
  {
    "name": "materiau"
  }
],
{
  "domainNetworkName": "RES_EAU",
  "source": "esriUNFCUTLine",
  "assetGroup": 100,
  "modifiableAttributes": [
    {
      "name": "corrosion",
      "replaceMode": 1
    }
  ]
}
]
```

## 3.2 ASSAINISSEMENT

### 3.2.1 General

La configuration générale de l'application « Assainissement » est définie dans le fichier « \config\onewater\sanitation\config.json ».

#### 3.2.1.1 Réseau

La configuration des informations du réseau est définie dans le groupe « **networkInfos** » :

```
"networkInfos": [
  {
    "label": "Réseau d'assainissement",
    "name": "sanitation",
    "webmap": "a3e5645f7fe146a0ae38bb32ae63cb8e",
    "webmap_mobile": "7201a1ba8b3b4c5782ec49983367d530",
    "networkService": "default:/1WATER_ORCL/1DevSanitation",
    "exploitationService": "default:/1WATER_ORCL/1OrclExploitation",
    "sectorizationService": "default:/1WATER_ORCL/1OrclSectorisation",
    "domainNetworkNames": [
      "RES_EP",
      "RES_EU"
    ]
  }
],
```

Figure 22 : Informations sur le réseau de distribution (Assainissement)

- ▶ **name** : Nom du réseau ;
- ▶ **label** : libellé qui s'affiche dans l'IHM ;
- ▶ **webmap** : identifiant de la carte ;
- ▶ **webmap\_mobile** : Si elle est remplie, cette clé désigne la carte à utiliser pour le mobile, si la clé est vide ou n'existe pas, la même carte est utilisée pour le web et le mobile
- ▶ **networkService** : URL d'accès aux services Utility Network ;
- ▶ **exploitationService** : URL d'accès aux services d'exploitation ;
- ▶ **sectorizationService** : URL d'accès aux services de sectorisation ;

► **domainNetworkNames** : liste du ou des domaine(s) du réseau de distribution.

### 3.2.1.2 Carte - Outil de localisation

La configuration de l'outil de localisation de l'application « Assainissement » est définie dans le groupe « **localisation** » de « **map** » :

```
"localisation": {
  "sources": [
    "BAN",
    {
      "featureServiceUrl": "default:/1WATER_ORCL/1SOrc1Sanitation",
      "layerId": 110,
      "searchFields": [
        "ident"
      ],
      "displayField": "ident",
      "outFields": [
        "ident"
      ],
      "name": "EP-Tronçon par identifiant",
      "filter": {
        "where": "assetgroup=200"
      }
    },
    {
      "featureServiceUrl": "default:/1WATER_ORCL/1SOrc1Sanitation",
      "layerId": 210,
      "searchFields": [
        "ident"
      ],
      "displayField": "ident",
      "outFields": [
        "ident"
      ],
      "name": "EU-Tronçon par identifiant",
      "filter": {
        "where": "assetgroup=300"
      }
    }
  ],
  "minSuggestCharacters": 3,
  "maxSuggestions": 5
},
```

Figure 23 : Localisation (Assainissement)

```
"localisation": {
  "sources": [
    {
      "name": "BAN",
      "options": {
        "postcode": 56900
      }
    },
    {
      "featureServiceUrl": "default:/1WATER_DEV/1DevSanitation",
      "layerId": 110,
      "searchFields": [
        "ident"
      ],
      "displayField": "ident",
      "outFields": [
        "ident"
      ],
      "name": "EP-Tronçon par identifiant",
      "filter": {
        "where": "assetgroup=200"
      }
    },
    {
      "featureServiceUrl": "default:/1WATER_DEV/1DevSanitation",
      "layerId": 210,
      "searchFields": [
        "ident"
      ],
      "displayField": "ident",
      "outFields": [
        "ident"
      ],
      "name": "EU-Tronçon par identifiant",
      "filter": {
        "where": "assetgroup=300"
      }
    }
  ],
  "minSuggestCharacters": 3,
  "maxSuggestions": 5
},
```

Figure 24 : Localisation - Restriction de la zone de recherche (Assainissement)

- **sources** : Listes des moteurs de recherche disponibles. Les valeurs de la liste peuvent être :
- « **BAN** » : Base d'adresse Nationale française. Ce type de moteur de recherche peut être configuré de deux manières :
    - Soit en renseignant la chaîne de caractères « **BAN** » (valeur par défaut)
    - Soit en renseignant un objet afin de restreindre le périmètre de recherche :
      - **name** : nom de la recherche = « **BAN** »
      - **options** : options permettant d'indiquer une commune.
        - **postcode** : code postal de la commune. Un seul code postal est possible
        - **citycode** : code INSEE de la commune. Un seul code INSEE est possible

Exemple :

```
{
  "name": "BAN",
  "options": {
    "postcode": 56800
  }
}
```

- « OSM » : Base d'adresse d'OpenStreetMap. Ce type de moteur de recherche peut être configuré de deux manières :
- Soit en renseignant la chaîne de caractères « OSM » (valeur par défaut)
- Soit en renseignant un objet afin de restreindre le périmètre de recherche :
- **name** : nom de la recherche = « OSM »
- **options** : options permettant de renseigner la zone :
- **viewcode** : cadrage correspondant à la zone de recherche, « x1, y1, x2, y2 ». Les valeurs sont à renseigner en WGS84 (x pour la longitude et y pour la latitude)

Exemple :

```
{  
  "name": "OSM",  
  "options": {  
    "viewcode": "-2.44846,47.86350,-2.30722,47.97302"  
  }  
}
```

- Objet décrivant les informations pour une recherche au niveau des objets d'une couche :
  - **featureServiceUrl** : Url vers le service de la couche ;
  - **layerId** : Identifiant de la couche (au niveau du service) ;
  - **searchFields** : La liste des noms de champs de la couche à rechercher ;
  - **displayField** : Nom du champ utilisé pour l'affichage des résultats. Par défaut, utilise le displayField indiqué au niveau de la couche ;
  - **outFields** : Spécifie les champs retournés avec les résultats de la recherche ;
  - **name** : Nom de la recherche proposée à l'utilisateur ;
  - **filter** : Filtre à appliquer lors de la recherche (where clause).
- **minSuggestCharacters** : Le nombre minimum de caractères à encoder avant que la recherche ne soit exécutée ;
- **maxSuggestions** : Le nombre maximum de valeurs proposées dans la liste.

### 3.2.1.3 Carte – Impression

La configuration de l'outil d'impression est définie dans le groupe « **print** » de « **map** ».

```
"map": {  
  "localisation": {  
    "print": {  
      "printServiceUrl": "default:/Utilities/PrintingTools/GPServer/Export%20Web%20Map%20Task",  
      "title": null,  
      "author": null,  
      "copyright": null,  
      "legendEnabled": true,  
      "scaleEnabled": false,  
      "allowedFormats": ["pdf", "jpg", "png32", "gif"],  
      "allowedLayouts": ["a4-portrait", "a4-landscape", "a3-portrait", "a3-landscape"],  
      "dpi": null,  
      "northArrowEnabled": false  
    }  
  }  
},
```

Figure 25 : Impression (Assainissement)

- **printServiceUrl** : Url vers le service d'impression.
- **title** : libellé à afficher comme titre par défaut de l'impression. Cette valeur est modifiable par l'utilisateur.

- **author** : libellé à renseigner comme auteur par défaut de l'impression. Si aucune valeur n'est fournie, le nom de l'utilisateur qui a demandé l'impression est affiché et utilisé dans l'application. Cette valeur est modifiable par l'utilisateur.
- **copyright** : libellé du copyright par défaut à afficher dans l'impression. Cette valeur est modifiable par l'utilisateur.
- **legendEnabled** : indique si une légende doit être affichée dans le résultat de l'impression. Cette valeur est modifiable par l'utilisateur.
- **scaleEnabled** : indique si l'impression se base sur l'échelle courante de la carte (false) ou sur la valeur de l'échelle renseignée dans le champ « échelle » dans l'application (true). Cette valeur est modifiable par l'utilisateur.
- **allowedFormats** : liste des formats d'impression disponibles.
- **allowedLayouts** : liste des canevas d'impression disponibles.
- **dpi** : résolution de la carte imprimée (exprimée en dots per inch). Si aucune valeur n'est fournie, la valeur par défaut (96 dpi) est affichée et utilisée dans l'application. Cette valeur est modifiable par l'utilisateur.
- **northArrowEnabled** : indique si l'orientation du nord doit être affichée.

### 3.2.1.4 Identifiants des couches

La liste des couches utilisées par le module est définie dans le groupe « **layers** » :

```
"layers": {  
  "operatorLayerId" : 20,  
  "anomalyLayerId": 202,  
  "blockageLayerId": 300,  
  "blockageHistoryId": 301,  
  "interventionLayerId": 400,  
  "interventionEquipmentLayerId": 401,  
  "interventionOperationLayerId": 402,  
  "interventionOperationItemLayerId": 403,  
  "interventionPointsLayerId": 404,  
  "tviLayerId": 500,  
  "tviDeviceLayerId": 501,  
  "tviDeviceAssetLayerId": 502,  
  "tviObservationLayerId": 503,  
  "tviDeviceNotation": 504,  
  "tviInspectionRubricLayerId": 505,  
  "tviObservationRubricLayerId": 506,  
  "tviPointsLayerId": 507,  
  "blackPointLayerId": 800,  
  "blackPointHistoryId": 801,  
  "scoringLayerId": 603,  
  "scoringPointsLayerId": 604,  
  "scoreLayerId" : 605,  
  "currentScoreLayerId" : 606,  
  "consequenceLayerId" : 1000,  
  "frameLayerId" : 1001,  
  "graffitiLayerId" : 1002  
},
```

*Figure 26 : Couches d'exploitation (Assainissement)*

- **operatorLayerId** : identifiant de la couche contenant le détail des différents opérateurs (véhicules et équipes) sur un curage (module des Curages) ;
- **anomalyLayerId** : identifiant de la couche qui représente les anomalies dans les services 1Water. Ce sont les objets de cette couche qui apparaissent dans la carte quand le module des anomalies est actif ;
- **blockageLayerId** : identifiant de la couche qui représente les encrassements dans les services 1Water. Ce sont les objets de cette couche qui apparaissent dans la carte quand le module des encrassements est actif ;

- ▶ **blockageHistoryId** : identifiant de la couche qui représente l'historique des encrassements dans les services 1Water. Les objets de cette couche n'apparaissent pas dans la carte. A 1 objet de la couche blockageLayerId peut correspondre 0 à n objets de la couche blockageHistoryId ;
- ▶ **interventionLayerId** : Identifiant de la couche des programmes d'intervention utilisée lors de l'affichage en mode normal (module « Curage » et « Visite ») ;
- ▶ **interventionEquipmentLayerId** : identifiant de la couche des interventions (module « Curage » et « Visite ») ;
- ▶ **interventionOperationLayerId** : identifiant de la couche des opérations d'intervention (module « Curage » et « Visite ») ;
- ▶ **interventionOperationItemLayerId** : identifiant la couche des éléments d'opération d'intervention. A chaque équipement, associé à une opération d'intervention, est associé un élément d'opération d'intervention (module « Curage » et « Visite ») ;
- ▶ **interventionPointsLayerId** : Identifiant de la couche des programmes d'intervention utilisée en mode cluster (module « Curage » et « Visite ») ;
- ▶ **tviLayerId** : Identifiant de la couche des programmes d'inspection (module « Inspection télévisée ») ;
- ▶ **tviDeviceLayerId** : Identifiant de la couche des ouvrages à inspecter (module « Inspection télévisée ») ;
- ▶ **tviDeviceAssetLayerId** : Identifiant de la couche des assets d'un ouvrage (module « Inspection télévisée ») ;
- ▶ **tviObservationLayerId** : Identifiant de la couche des observations (module « Inspection télévisée ») ;
- ▶ **tviDeviceNotation** : Identifiant de la couche des notations (module « Inspection télévisée »).
- ▶ **tviInspectionRubricLayerId** : Identifiant de la couche des rubriques d'inspection (module « Inspection télévisée ») ;
- ▶ **tviObservationRubricLayerId** : Identifiant de la couche des rubriques d'observation (module « Inspection télévisée ») ;
- ▶ **tviPointsLayerId** : Identifiant de la couche des programmes d'inspection utilisée en mode cluster (module « Inspection télévisée ») ;
- ▶ **blackPointLayerId** : identifiant de la couche qui représente les points noirs dans les services 1Water. Ce sont les objets de cette couche qui apparaissent dans la carte quand le module des points noirs est actif ;
- ▶ **blackPointHistoryId** : identifiant de la couche qui représente l'historique des points noirs dans les services 1Water. Les objets de cette couche n'apparaissent pas dans la carte. A 1 objet de la couche blackPointLayerId peut correspondre 0 à n objets de la couche blackPointHistoryId ;
- ▶ **scoringLayerId** : identifiant de la couche qui représente un projet de notation dans les services 1Water. Ce sont les objets de cette couche qui apparaissent dans la carte lorsque le module des notations est actif ;
- ▶ **scoringPointsLayerId** : identifiant de la couche qui représente le centre géographique d'un projet de notation dans les services 1Water. Ce sont les objets de cette couche qui apparaissent dans la carte sous forme de pins lorsque le module des notations est actif ;
- ▶ **scoreLayerId** : identifiant de la couche qui représente une note calculée dans les services 1Water. Une note se décompose en notes et/ou critères ;
- ▶ **currentScoreLayerId** : identifiant de la couche qui représente une note calculée et validée dans les services 1Water. Une note se décompose en notes et/ou critères.
- ▶ **frameLayerId** : identifiant de la couche qui représente les cadrages ou les emplacements d'une zone de la carte. Ces cadrages sont enregistrés qu'en mobilité.
- ▶ **consequenceLayerId** : identifiant de la couche de gestion des différences observées au niveau du patrimoine entre les valeurs de la base de données et les valeurs constatées sur le terrain ou au travers des inspections télévisées.

- **graffitiLayerId** : Identifiant de la couche qui représente les graffitis ou annotations graphiques géolocalisés.

La liste des couches de sectorisation, utilisées par le module pour le calcul des notes, est définie dans le groupe « **sectorizationlayers** » :

```
"sectorizationlayers": {  
  "scoreGeoZone": 600,  
  "LevelScoreGeoZone": 601  
}
```

Figure 27 : Couches de sectorisation (Assainissement)

- **scoreGeoZone**: identifiant de la couche de zonage pour le calcul des notes ;
- **LevelScoreGeoZone** : identifiant de la couche de zonage par niveau pour le calcul des notes.

La liste des couches de dépose utilisées par le module est définie dans le groupe « **businessdatalayers** ».

L'application d'assainissement propose deux modèles de données pour la gestion du patrimoine :

- Le modèle à un domaine regroupant les eaux usées (unitaire) et les eaux pluviales.
- Le modèle à deux domaines pour une gestion en séparatif strict entre les eaux usées (unitaire) et les eaux pluviales.

La distinction entre les couches métiers de l'assainissement s'appuie sur le nom du domaine en préfixe.

Pour l'assainissement avec deux domaines :

```
"businessdatalayers": {  
  "RES_EPRemovalZone": 2100,  
  "RES_EPRemovalPipe": 2110,  
  "RES_EPRemovalJunction": 2120,  
  "RES_EPRemovalDevice": 2130,  
  "RES_EPRemovalAssembly": 2140,  
  "RES_EURemovalZone": 2200,  
  "RES_EURemovalPipe": 2210,  
  "RES_EURemovalJunction": 2220,  
  "RES_EURemovalDevice": 2230,  
  "RES_EURemovalAssembly": 2240  
}
```

Pour l'assainissement avec un domaine :

```
"businessdatalayers": {  
  "RES_ASSRemovalZone": 2100,  
  "RES_ASSRemovalPipe": 2110,  
  "RES_ASSRemovalJunction": 2120,  
  "RES_ASSRemovalDevice": 2130,  
  "RES_ASSRemovalAssembly": 2140  
}
```

- **<Nom de domaine>RemovalZone** : identifiant de la couche de zonage des déposes du domaine concerné ;
- **<Nom de domaine>RemovalPipe** : identifiant de la couche de linéaire déposé du domaine concerné ;
- **<Nom de domaine>RemovalJunction** : identifiant de la couche de jonction déposé du domaine concerné ;
- **<Nom de domaine>RemovalDevice** : identifiant de la couche d'appareil déposé du domaine concerné ;

- **<Nom de domaine>RemovalAssembly** : identifiant de la couche d'assemblage déposé du domaine concerné.

### 3.2.1.5 Rapports

La configuration des rapports est définie dans le groupe « **reports** » :

```
"reports": {  
  "tvi": {  
    "filepath": "TVI_SYNTHESIS"  
  },  
  "tviDevice": {  
    "filepath": "TVI_DEVICE"  
  },  
  "tviDeviceRerau": {  
    "filepath": "TVI_DEVICE_RERAU"  
  },  
  "tviAllDevices" : {  
    "filepath": "TVI_ALLDEVICES"  
  },  
  "tviAllDevicesRerau": {  
    "filepath": "TVI_ALLDEVICES_RERAU"  
  },  
  "cleaning" : {  
    "filepath": "CLEANING_SYNTHESIS"  
  },  
  "cleaningOperation" : {  
    "filepath": "CLEANING_OPERATION"  
  },  
  "visit": {  
    "filepath": "VISIT_SYNTHESIS"  
  },  
  "visitOperation": {  
    "filepath": "VISIT_OPERATION"  
  },  
  "deviceSynthesis": {  
    "filepath": "EQPT_ASS",  
    "maxScale": 2000  
  }  
}
```

**Figure 28 : Rapports (Assainissement)**

- **tvi** : Génère une synthèse d'une inspection télévisée.
  - **filepath** : Nom du fichier qui permet de générer le rapport précédé du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.
- **tviAllDevices** : Génère un rapport détaillé d'une inspection télévisée, basé sur la méthode de calcul des notes « STANDARD ».
  - **filepath** : Nom du fichier qui permet de générer le rapport précédé du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.
- **tviDevice** : Génère un rapport détaillé sur un ouvrage spécifique d'une inspection télévisée, basé sur la méthode de calcul des notes « STANDARD ».
  - **filepath** : Nom du fichier qui permet de générer le rapport précédé du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.
- **tviAllDevicesRerau** : Génère un rapport détaillé d'une inspection télévisée, basé sur la méthode de calcul des notes « RERAU ».
  - **filepath** : Nom du fichier qui permet de générer le rapport précédé du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.
- **tviDeviceRerau** : Génère un rapport détaillé sur un ouvrage spécifique d'une inspection télévisée, basé sur la méthode de calcul des notes « RERAU ».
  - **filepath** : Nom du fichier qui permet de générer le rapport précédé du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.
- **cleaning** : Génère un rapport détaillé d'un curage.

- **filepath** : Nom du fichier qui permet de générer le rapport précédé du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.
- ▶ **cleaningOperation** : Génère un rapport détaillé pour une date d'opérations spécifique d'un curage.
- ▶ **filepath** : Nom du fichier qui permet de générer le rapport précédé du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.
- ▶ **visit** : Génère un rapport détaillé d'une visite.
- ▶ **filepath** : Nom du fichier qui permet de générer le rapport précédé du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.
- ▶ **visitOperation** : Génère un rapport détaillé pour une date d'opérations spécifique d'une visite.
  - **filepath** : Nom du fichier qui permet de générer le rapport précédé du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.
- ▶ **deviceSynthesis** : Génère une fiche de vie de l'équipement.
  - **filepath** : Nom du fichier qui permet de générer le rapport précédé du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.
  - **maxScale** : Echelle maximale de cadrage pour le plan de situation associé au rapport.

### 3.2.1.6 Service d'impression

La configuration du service d'impression de l'application « Assainissement » est définie dans le groupe « **printService** » :

```
"printService": {  
  "url": "default:/Utilities/PrintingTools/GPServer/Export%20Web%20Map%20Task"  
},
```

*Figure 29 : Service d'impression (Assainissement)*

- ▶ **url** : Url vers le service d'impression ;

### 3.2.1.7 Tâches asynchrones

#### 3.2.1.7.1 Tâche asynchrone de gestion des suppressions (dépose)

Une tâche planifiée asynchrone pour le métier « Assainissement » permet à la suite de la suppression, dans le cadre ou non d'une dépose, d'un ou plusieurs éléments (tronçon, appareil, ...), de mettre à jour automatiquement l'état des anomalies, des encrassements et des points noirs qui sont liées au(x) élément(s) supprimé(s).

La configuration de cette tâche est définie dans le groupe « **parameters** » :

```
"parameters": {  
  "syncRemovalCron": "0 0 23 * * *",  
  "syncCsqCron": "0 0 21 * * *",  
  "maxUploadAttachmentSize": 5  
},
```

- ▶ **syncRemovalCron** : fréquence d'exécution de la tâche. La syntaxe de ce paramètre correspond au principe du cron UNIX/LINUX (seconde 0-59, minute 0-59, heure 0-23, jour du mois 1-31, mois 1-12, jour de la semaine 0 pour dimanche à 6 pour samedi).

**Exemples :**



- 0 0 23 \* \* \* : chaque jour à 23 heures 00 minutes 00 secondes ;
- 0 0 \* \* \* \* : chaque heure de chaque jour ;
- 0 0 8-10 \* \* \* : 8, 9 et 10 heures chaque jour à 8,9 et 10 heures ;
- 0 0 6,19 \* \* \* : chaque jour à 6 heures et à 19 heures ;
- 0 0/30 8-10 \* \* \* : chaque jour à 8:00, 8:30, 9:00, 9:30, 10:00 et 10:30 ;
- 0 0 9-17 \* \* MON-FRI : toutes les heures entre 9 heures et 17 heures du lundi au vendredi ;
- 0 0 0 25 12 ? : chaque 25 décembre à minuit ;
- 0 0 0 L \* \* : dernier jour du mois à minuit ;
- 0 0 0 \* \* THUL : dernier jeudi du mois à minuit ;
- 0 0 0 ? \* 5#2 : deuxième vendredi du mois à minuit.

! Si la clé **syncRemovalCron** n'est pas présente la tâche asynchrone n'est pas déclenchée.

### 3.2.1.7.2 Tâche asynchrone de la remontée des conséquences

Certains modules permettent de corriger la valeur d'attributs du patrimoine de l'assainissement. Pour des raisons de cohérence et d'efficacité, ces corrections ne sont pas réalisées directement mais par l'intermédiaire d'une tâche asynchrone.

Ce mécanisme est utilisé pour la remontée d'information attributaire des inspections télévisées.

La configuration de cette tâche est définie dans le groupe « **parameters** » :

```
"parameters": {  
  "syncRemovalCron": "0 0 23 * * *",  
  "syncCsqCron": "0 0 21 * * *",  
  "maxUploadAttachmentSize": 5  
},
```

- **syncCsqCron** : fréquence d'exécution de la tâche. La syntaxe de ce paramètre correspond au principe du cron UNIX/LINUX (seconde 0-59, minute 0-59, heure 0-23, jour du mois 1-31, mois 1-12, jour de la semaine 0 pour dimanche à 6 pour samedi).

#### Exemples :

- 0 0 23 \* \* \* : chaque jour à 23 heures 00 minutes 00 secondes ;
- 0 0 \* \* \* \* : chaque heure de chaque jour ;
- 0 0 8-10 \* \* \* : 8, 9 et 10 heures chaque jour à 8,9 et 10 heures ;
- 0 0 6,19 \* \* \* : chaque jour à 6 heures et à 19 heures ;
- 0 0/30 8-10 \* \* \* : chaque jour à 8:00, 8:30, 9:00, 9:30, 10:00 et 10:30 ;
- 0 0 9-17 \* \* MON-FRI : toutes les heures entre 9 heures et 17 heures du lundi au vendredi ;
- 0 0 0 25 12 ? : chaque 25 décembre à minuit ;
- 0 0 0 L \* \* : dernier jour du mois à minuit ;
- 0 0 0 \* \* THUL : dernier jeudi du mois à minuit ;
- 0 0 0 ? \* 5#2 : deuxième vendredi du mois à minuit.

! Si la clé **syncCsqCron** n'est pas présente la tâche asynchrone n'est pas déclenchée.

### 3.2.1.8 Pièces jointes

La configuration des pièces jointes est définie dans le groupe « **parameters** » :

```
"parameters": {  
  "syncRemovalCron": "0 0 23 * * *",  
  "syncCsqCron": "0 0 21 * * *",  
  "maxUploadAttachmentSize": 5  
},
```

**maxUploadAttachmentSize** : Taille maximum autorisée d'un fichier à télécharger en tant que pièce jointe. Cette valeur est à renseigner en Mégabytes.

! Les pièces jointes sont copiées dans la base et impacte donc sa volumétrie.

### 3.2.1.9 Modèle numérique de terrain

Le modèle numérique de terrain est exploité par le webservice de calcul d'altitude et par le profil en long.

La configuration du modèle numérique de terrain est définie dans le groupe « **mnt** » :

```
"mnt": {  
  "accessor": "gdal",  
  "source": "gdal001/morbihan.vrt"  
}
```

*Figure 30 : Modèle numérique de terrain (Assainissement)*

- **accessor** : indique le logiciel d'accès aux données du MNT (uniquement gdal dans cette version) ;
- **source** : indique le chemin du fichier « .vrt ». Le chemin est relatif, il s'appuie sur le répertoire « <MonServeur>\config\permanent\mnt ».

! Les données MNT doivent être dans la même référence spatiale que les données ESRI.

## 3.2.2 Anomalie

Ce module permet de créer, modifier ou supprimer une anomalie, de renseigner les informations de cette dernière ainsi que les informations relatives à l'intervention qui a été effectuée pour la réparer.

Le module « Anomalies » est associé à des droits fonctionnels (consultation, édition et gestion). Il est nécessaire de posséder au moins le droit de consultation pour accéder à ce module.

L'utilisateur peut également être associé à une restriction géographique pour contrôler ses actions d'édition et de gestion.

### 3.2.2.1 Configuration

La configuration du module « Anomalie » est définie dans le fichier « \config\onewater\sanitation\anomaly.json » :

```
"maxItemsInList": 50,  
"networkEntities": [  
  "filters": [  
    {
```

*Figure 31 : Anomalie (Assainissement)*

- **maxItemsInList** : nombre maximum d'anomalies qu'il est possible d'afficher dans la carte et dans la liste des anomalies. Si dans l'étendue courante de la carte il y a plus d'anomalies que la valeur de « maxItemsInList » :
  - L'affichage des anomalies dans la carte bascule en mode « cluster » ;
  - La liste contient un message qui indique notamment le nombre d'anomalies présentes (en lieu et place de la liste des anomalies présentes dans la carte).
- ! Plus la valeur de ce paramètre est élevée, plus la fluidité d'affichage de la carte risque d'être dégradée.
- **networkEntities** : Informations sur les types d'équipements qui peuvent faire l'objet d'une anomalie. Pour chaque type :
  - **source** : type d'usage de l'équipement au niveau du réseau utility network.  
Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction » ;
  - **assetGroup** : code de l'assetGroup correspondant à l'équipement au sein de la couche concernée ;
  - **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné ;
  - **typeRestrictions** : liste des codes correspondants aux types d'anomalie à proposer pour ce type d'équipement (filtre qui permet de faire une dépendance entre le type d'équipement choisi et les types d'anomalies proposés pour ce type d'équipement).

```
"networkEntities": [  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 200,  
    "domainNetworkName": "RES_EP",  
    "typeRestrictions": [  
      1,  
      2,  
      5,  
      6,  
      7,  
      99  
    ]  
  },  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 202,  
    "domainNetworkName": "RES_EP",  
    "typeRestrictions": [  
      2  
    ]  
  },  
  {  
    "source": "esriUNFCUTJunction",  
    "assetGroup": 220,  
    "domainNetworkName": "RES_EP",  
    "typeRestrictions": [  
      8  
    ]  
  },  
  {  
    "source": "esriUNFCUTJunction",  
    "assetGroup": 221,  
    "domainNetworkName": "RES_EP",  
    "typeRestrictions": [  
      8  
    ]  
  },  
  {  
    "source": "esriUNFCUTJunction".
```

Figure 32 : Patrimoine pour anomalie (Assainissement)

! Si cette liste est mal configurée (vide ou aucune valeur correspondant au modèle mis en place), le module sera uniquement accessible en consultation

► **filters** : liste des filtres de sélection personnalisés. Un filtre de sélection personnalisé permet d'affiner la portée de la sélection qui est faite dans la carte. Au lieu de sélectionner tous les équipements « sélectionnables », on ne va sélectionner que ceux qui répondent à la condition liée au filtre personnalisé actif.

Par exemple :

- Uniquement les équipements ponctuels
- Uniquement les regards
- Uniquement les équipements qui sont dans le domaine « eau usée »
- Uniquement les tronçons ou les regards qui sont dans le domaine « eau usée »
- Uniquement les équipements qui sont visitables
- Uniquement les tronçons qui sont en PVC, en acier ou en fonte et qui ont une longueur supérieure à 40m
- ...

Des exemples de filtres personnalisés sont disponibles en [annexe](#).

Pour chaque filtre personnalisé :

- **name** : nom du filtre (obligatoire).
- **description** : description du filtre. Optionnel mais recommandé. Permet à l'utilisateur de savoir quelle est la portée du filtre et la(les) condition(s) appliquée(s)
- **condition** : condition à appliquer sur les équipements du réseaux lors de la sélection
  - **networkEntities** : informations sur les types d'équipements qui peuvent être sélectionnés. Doit être un sous-ensemble du *networkEntities* déclaré au niveau supérieur. Optionnel.  
Pour chaque type :
- **source** : type d'usage de l'équipement au niveau du réseau utility network.  
Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction » ;
- **assetGroup** : code de l'assetGroup correspondant à l'équipement au sein de la couche concernée ;
- **assetType** : code de l'assetType correspondant au type d'équipement de l'assetGroup concerné. Exemple : tronçon (= assetGroup) de type refoulement, décharge, simple, etc... (=assetType) ;
- **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné ;
  - **whereClause** : condition qui s'applique soit sur les types d'équipements renseignés dans networkEntities (si networkEntities est renseigné) soit sur tous les types d'équipements sélectionnables pour le module (cfr. networkEntities du niveau supérieur).

### 3.2.3 Encrassement

Le module permet de créer, de modifier ou de supprimer un encrassement.

Le module « Encrassements » est associé à des droits fonctionnels (consultation, édition et gestion).

L'utilisateur peut également être associé à une restriction géographique pour contrôler ses actions d'édition et de gestion.

### 3.2.3.1 Configuration

La configuration du module « Encrassement » est définie dans le fichier « \config\onewater\sanitation\blockage.json ».

```
"maxItemsInList": 50,  
"networkEntities": [  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 200,  
    "domainNetworkName": "RES_ASS"  
  },  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 201,  
    "domainNetworkName": "RES_ASS"  
  },  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 203,  
    "domainNetworkName": "RES_ASS"  
  },  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 204,  
    "domainNetworkName": "RES_ASS"  
  },  
  {  
    "source": "esriUNFCUTDevice",  
    "assetGroup": 263,  
    "domainNetworkName": "RES_ASS"  
  }  
],  
"filters": [  
  {  
    "name": "Eaux pluviales",  
    "description": "Uniquement les équipements du domaine des eaux pluviales",  
    "condition": {  
      "whereClause": "reseau = 1"  
    }  
  },  
  {  
    "name": "Eaux usées",  
    "description": "Uniquement les équipements du domaine des eaux usées",  
    "condition": {  
      "whereClause": "reseau = 2"  
    }  
  },  
  {  
    "name": "Petites canalisations",  
    "description": "Équipements de type linéaires inférieurs à 15m uniquement",  
    "condition": {  
      "whereClause": "longueur < 15",  
      "networkEntities": [  
        { "source": "esriUNFCUTLine", "assetGroup": 200, "domainNetworkName": "RES_ASS" },  
        { "source": "esriUNFCUTLine", "assetGroup": 201, "domainNetworkName": "RES_ASS" },  
        { "source": "esriUNFCUTLine", "assetGroup": 203, "domainNetworkName": "RES_ASS" },  
        { "source": "esriUNFCUTLine", "assetGroup": 204, "domainNetworkName": "RES_ASS" }  
      ]  
    }  
  }  
]
```

Figure 33 : Encrassement

► **maxItemsInList** : nombre maximum d'encrassements qu'il est possible d'afficher dans la carte et dans la liste des encrassements. Si dans l'étendue courante de la carte il y a plus d'encrassements que la valeur de « maxItemsInList » :

- L'affichage des encrassements dans la carte bascule en mode « cluster » ;
- La liste des encrassements contient un message qui indique notamment le nombre d'encrassements présents (en lieu et place de la liste des encrassements présents dans la carte).

! Plus la valeur de ce paramètre est élevée, plus la fluidité d'affichage de la carte risque d'être dégradée.

► **networkEntities** : informations sur les types d'équipements qui peuvent faire l'objet d'un encrassement. Pour chaque type :

- **source** : type d'usage de l'équipement au niveau du réseau utility network.  
Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction » ;
- **assetGroup** : code de l'assetGroup correspondant à l'équipement au sein de la couche concernée ;
- **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné.

! Si cette liste est mal configurée (vide ou aucune valeur correspondant au modèle de données), le module est uniquement accessible en consultation.

► **filters** : liste des filtres de sélection personnalisés. Un filtre de sélection personnalisé permet d'affiner la portée de la sélection qui est faite dans la carte. Au lieu de sélectionner tous les équipements « sélectionnables », on ne va sélectionner que ceux qui répondent à la condition liée au filtre personnalisé actif.

Par exemple :

- Uniquement les équipements linéaires
- Uniquement les regards
- Uniquement les équipements qui sont dans le domaine « eau usée »
- Uniquement les tronçons ou les regards qui sont dans le domaine « eau usée »
- Uniquement les équipements qui sont visitables
- Uniquement les tronçons qui sont en PVC, en acier ou en fonte et qui ont une longueur supérieure à 40m
- ...

Des exemples de filtres personnalisés sont disponibles en [annexe](#).

Pour chaque filtre personnalisé :

- **name** : nom du filtre (obligatoire).
- **description** : description du filtre. Optionnel mais recommandé. Permet à l'utilisateur de savoir quelle est la portée du filtre et la(les) condition(s) appliquée(s)
- **condition** : condition à appliquer sur les équipements du réseaux lors de la sélection
  - **networkEntities** : informations sur les types d'équipements qui peuvent être sélectionnés. Doit être un sous-ensemble du *networkEntities* déclaré au niveau supérieur. Optionnel.  
Pour chaque type :
    - **source** : type d'usage de l'équipement au niveau du réseau utility network.  
Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction » ;
    - **assetGroup** : code de l'assetGroup correspondant à l'équipement au sein de la couche concernée ;
    - **assetType** : code de l'assetType correspondant au type d'équipement de l'assetGroup concerné. Exemple : tronçon (= assetGroup) de type refoulement, décharge, simple, etc... (=assetType) ;

- **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné ;
  - **whereClause** : condition qui s'applique soit sur les types d'équipements renseignés dans networkEntities (si networkEntities est renseigné) soit sur tous les types d'équipements sélectionnables pour le module (cfr. networkEntities du niveau supérieur).

### 3.2.4 Point noir

Le module permet de créer, de modifier ou de supprimer un point noir.

Le module « Point noir » est associé à des droits fonctionnels (consultation, édition et gestion).

L'utilisateur peut également être associé à une restriction géographique pour contrôler ses actions d'édition et de gestion.

#### 3.2.4.1 Configuration

La configuration du module « Point noir » est définie dans le fichier « \config\onewater\sanitation\blackpoint.json » :

```
"maxItemsInList": 50,  
"networkEntities": [  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 200,  
    "domainNetworkName": "RES_ASS"  
  },  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 201,  
    "domainNetworkName": "RES_ASS"  
  },  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 203,  
    "domainNetworkName": "RES_ASS"  
  },  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 204,  
    "domainNetworkName": "RES_ASS"  
  },  
  {  
    "source": "esriUNFCUTDevice",  
    "assetGroup": 263,  
    "domainNetworkName": "RES_ASS"  
  }  
],  
"filters": [  
  {  
    "name": "Eaux pluviales",  
    "description": "Uniquement les équipements du domaine des eaux pluviales",  
    "condition": {  
      "whereClause": "reseau = 1"  
    }  
  }  
],  
"automatic": {  
  "periodInYears": 1,  
  "minCleaningsInPeriod": 2  
}
```

Figure 34 : Point noir

- **maxItemsInList** : nombre maximum de points noirs qu'il est possible d'afficher dans la carte et dans la liste des points noirs. Si dans l'étendue courante de la carte il y a plus de points noirs que la valeur de « maxItemsInList » :
  - L'affichage des points noirs dans la carte bascule en mode « cluster » ;
  - La liste des points noirs contient un message qui indique notamment le nombre de points noirs présents (en lieu et place de la liste des points noirs présents dans la carte).

! Plus la valeur de ce paramètre est élevée, plus la fluidité d’affichage de la carte risque d’être dégradée

► **networkEntities** : informations sur les types d’équipements qui peuvent faire l’objet d’un point noir. Pour chaque type :

- **source** : type d’usage de l’équipement au niveau du réseau utility network.  
Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction » ;
- **assetGroup** : code de l’assetGroup correspondant à l’équipement au sein de la couche concernée ;
- **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d’équipement concerné.

! Si cette liste est mal configurée (vide ou aucune valeur correspondant au modèle de données), le module est uniquement accessible en consultation.

► **filters** : liste des filtres de sélection personnalisés. Un filtre de sélection personnalisé permet d’affiner la portée de la sélection qui est faite dans la carte. Au lieu de sélectionner tous les équipements « sélectionnables », on ne va sélectionner que ceux qui répondent à la condition liée au filtre personnalisé actif.

Par exemple :

- Uniquement les équipements linéaires
- Uniquement les regards
- Uniquement les équipements qui sont dans le domaine « eau usée »
- Uniquement les tronçons ou les regards qui sont dans le domaine « eau usée »
- Uniquement les équipements qui sont visitables
- Uniquement les tronçons qui sont en PVC, en acier ou en fonte et qui ont une longueur supérieure à 40m
- ...

Des exemples de filtres personnalisés sont disponibles en [annexe](#).

Pour chaque filtre personnalisé :

- **name** : nom du filtre (obligatoire).
- **description** : description du filtre. Optionnel mais recommandé. Permet à l’utilisateur de savoir quelle est la portée du filtre et la(les) condition(s) appliquée(s)
- **condition** : condition à appliquer sur les équipements du réseaux lors de la sélection
  - **networkEntities** : informations sur les types d’équipements qui peuvent être sélectionnés. Doit être un sous-ensemble du *networkEntities* déclaré au niveau supérieur. Optionnel.  
Pour chaque type :
    - **source** : type d’usage de l’équipement au niveau du réseau utility network.  
Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction » ;
    - **assetGroup** : code de l’assetGroup correspondant à l’équipement au sein de la couche concernée ;



- **assetType** : code de l'assetType correspondant au type d'équipement de l'assetGrout concerné. Exemple : tronçon (= assetGroup) de type refoulement, décharge, simple, etc... (=assetType) ;
- **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné ;
  - **whereClause** : condition qui s'applique soit sur les types d'équipements renseignés dans networkEntities (si networkEntities est renseigné) soit sur tous les types d'équipements sélectionnables pour le module (cfr. networkEntities du niveau supérieur).
- **automatic** : critères pour le calcul automatique des points noirs
  - **periodInYears** : nombre d'année(s), à prendre en compte lors du calcul automatique, précédent une date de fin donnée ;
  - **minCleaningsInPeriod** : nombre minimum de curage(s) clôturé(s) sur un équipement, pendant la période définie (date – **periodInYears**), pour qu'un point noir calculé (automatique) soit créé.

## 3.2.5 Export de données

### 3.2.5.1 Configuration d'export de données

La configuration du module « Export de données » est définie dans le fichier « \config\onewater\sanitation\export.json » au niveau de « **services** » :

```
{
  "services": [
    {
      "name": "SWMU",
      "url": "default:/{{{sanitation.exportServices.SWMUUrl}}}"
    },
    {
      "name": "STARDT",
      "url": "default:/{{{sanitation.exportServices.STARDTUrl}}}"
    }
  ]
}
```

**Figure 35 : Export (Assainissement)**

Chaque type d'export est défini par :

- **nom** : Nom associé au format d'export ;
- **url** : Url à appeler pour réaliser le traitement de conversion vers le format attendu.

## 3.2.6 Propagation

Ce module permet de réaliser des propagations en amont ou en aval sur le réseau d'assainissement en partant d'un point de départ.

### 3.2.6.1 Configuration de la trace

Une propagation en amont ou en aval est exécutée par « Utility Network » qui effectue une trace à partir d'un élément du réseau.

La configuration s'appuie sur des traces nommées, deux traces par domaine d' « Utility network », qui exploitent des catégories spécifiques (PROPAGATION\_LINEAR, PROPAGATION\_IN et PROPAGATION\_OUT) pour préciser le résultat de la propagation.

La configuration du module « Propagation » est définie dans le fichier « \config\onewater\sanitation\propagation.json » :

```
{
  "traceConfiguration": {
    "upStream": {
      "names": [
        "1W_SW_UPSTREAM",
        "1W_WW_UPSTREAM"
      ],
      "defaultNames": [
        "1W_SW_UPSTREAM",
        "1W_WW_UPSTREAM"
      ]
    },
    "downStream": {
      "names": [
        "1W_SW_DOWNSTREAM",
        "1W_WW_DOWNSTREAM"
      ],
      "defaultNames": [
        "1W_SW_DOWNSTREAM",
        "1W_WW_DOWNSTREAM"
      ]
    }
  }
}
```

Figure 36 : Propagation (Assainissement 2 domaines)

```
{
  "traceConfiguration": {
    "upStream": {
      "defaultNames": [
        "1W_S_UPSTREAM"
      ],
      "defaultNames": [
        "1W_S_UPSTREAM"
      ]
    },
    "downStream": {
      "defaultNames": [
        "1W_S_DOWNSTREAM"
      ],
      "names": [
        "1W_S_DOWNSTREAM"
      ]
    }
  }
}
```

Figure 37 : Propagation (Assainissement 1 domaine)

► **upStream :**

- **names** : liste du (des) nom(s) de trace nommée pour une propagation en amont (une ou plusieurs par domaine).
- **defaultNames** : nom de la trace nommée par défaut pour une propagation en amont pour chaque domaine.

► **downStream :**

- **names** : liste du (des) nom(s) de trace nommée pour une propagation en aval (une ou plusieurs par domaine).
- **defaultNames** : nom de la trace nommée par défaut pour une propagation en aval pour chaque domaine.

! Toutes les configurations de trace pour le module « Propagation » doivent être associées à la balise « PROPAGATION » pour être prises en compte dans l'application de configuration.

### 3.2.7 Profil en long

Ce module permet de réaliser des profils en long sur le réseau d'assainissement.

La configuration du module « Profil en long » est définie dans le fichier « \config\onewater\sanitation\longprofile.json ».

Pour plus de détails, consulter le chapitre « [Profil en long](#) » dans la partie « Annexes ».

## 3.2.8 Inspection télévisée

Le module permet de préparer une inspection télévisée (ITV) et d'exploiter le fichier résultat produit par une campagne d'inspection.

Le module « Inspection télévisée » est associé à des droits fonctionnels (consultation, édition et gestion).

! La version courante de 1Water gère uniquement les branchements et les collecteurs de la norme EN-13508-2+A1 et leurs codes d'observation principaux associés (BA, BB, BC et BD).

### 3.2.8.1 Configuration

La configuration du module « Inspection télévisée » est définie dans le fichier « \config\onewater\sanitation\tvi.json ».

#### 3.2.8.1.1 Configuration générale

Cette section permet de définir les paramètres liés à l'affichage, à la génération du fichier canevas de la norme EN 13508-2, à la méthode de notation et au tri des ouvrages à inspecter :

```
"buffer": 5.0,  
"maxItemsInList": 50,  
"fileCharSet": "ISO-8859-1:1998",  
"fileLanguage": "fr",  
"fileFieldSeparator": ";",  
"fileDecimalDelimiter": ".",  
"fileTextDelimiter": " ",  
"fileVersion": "2010",  
"exportStandardName": "EN13508-2+A1-2011-08",  
"scoreMethod": 0,  
"photosRequired": true,  
"videosRequired": true,  
"templateCustomSortField": ""
```

Figure 38 : Généralités ITV

- **buffer** : Taille en mètre de la zone tampon autour des ouvrages à inspecter ;
- **maxItemsInList** : nombre maximum d'inspections télévisées qu'il est possible d'afficher dans la carte et dans la liste des inspections. Si dans l'étendue courante de la carte il y a plus d'inspections que la valeur de « maxItemsInList » :
  - L'affichage des inspections dans la carte bascule en mode « cluster » ;
  - La liste des inspections télévisées contient un message qui indique notamment le nombre d'inspections présentes (en lieu et place de la liste des inspections présentes dans la carte).
- ! Plus la valeur de ce paramètre est élevée, plus la fluidité d'affichage de la carte risque d'être dégradée.
- **fileCharSet** : Nom de l'encodage du fichier canevas ;
- **fileLanguage** : Code de langue qui sera indiqué dans le fichier canevas ;
- **fileFieldSeparator** : Délimiteur des champs dans le fichier canevas (uniquement pour le format txt) ;
- **fileTextDelimiter** : Délimiteur des chaînes dans le fichier canevas (uniquement pour le format txt) ;
- **fileVersion** : Numéro de version indiqué dans le fichier canevas ;
- **exportStandardName** : Nom de la norme utilisée pour l'export. Les valeurs possibles sont « EN13508-2+A1-2011-08 », « BEFDSS\_01\_01 » et « BEFDSS\_02\_01 ».

- ▶ **scoreMethod** : Méthode de notation à appliquer STANDARD (0) ou RERAU (1) ;
- ▶ **photosRequired** : Valeur par défaut : « true ». Lors de l'import d'un fichier résultat, si une ou plusieurs photos, référencées dans les rubriques d'observation, sont introuvables dans le dossier résultat :
  - l'import est interrompu si « photosRequired » vaut « true » ;
  - l'import est réalisé si « photosRequired » vaut « false » ;
- ▶ **videosRequired** : Valeur par défaut : « true ». Lors de l'import d'un fichier résultat, si une ou plusieurs vidéos, référencées dans les rubriques d'inspection, sont introuvables dans le dossier résultat :
  - l'import est interrompu si « videosRequired » vaut « true » ;
  - l'import est réalisé si « videosRequired » vaut « false » ;
- ▶ **templateCustomSortField** : Nom d'un champ des objets du patrimoine utilisé pour trier les équipements dans le fichier canevas (numéro de contrat du prestataire par exemple). Ce champ doit obligatoirement exister sur toutes les couches concernées par une inspection télévisée.

#### 3.2.8.1.2 Configuration du patrimoine

La section « **networkEntities** » permet de définir la liste des éléments du patrimoine concernés par une inspection télévisée :

```
"networkEntities": [  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 300,  
    "domainNetworkName": "RES_EU",  
    "nodes": [  
      {  
        "domainNetworkName": "RES_EU",  
        "source": "esriUNFCUTDevice",  
        "assetGroup": 363  
      }  
    ]  
  },  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 302,  
    "domainNetworkName": "RES_EU",  
    "nodes": [  
      {  
        "domainNetworkName": "RES_EU",  
        "source": "esriUNFCUTDevice",  
        "assetGroup": 361  
      },  
      {  
        "domainNetworkName": "RES_EU",  
        "source": "esriUNFCUTDevice",  
        "assetGroup": 363  
      },  
      {  
        "domainNetworkName": "RES_EU",  
        "source": "esriUNFCUTJunction",  
        "assetGroup": 324  
      }  
    ]  
  }  
],
```

Figure 39 : Patrimoine pour les inspections du réseau

Chaque bloc est composé des informations d'un linéaire éligible pour créer une nouvelle ITV, suivi d'un champ « **nodes** » qui liste les informations sur les différents nœuds possibles qu'on peut connecter à ce linéaire. Chaque élément nœud doit être associé au niveau du modèle de données à la catégorie **ITV\_ACCES** pour être pris en compte.

Cette liste des éléments du patrimoine est également utilisée lors de l'association avec un équipement d'un programme d'inspection qui n'a pas été rattaché automatiquement lors de l'import du fichier résultat.

! Si cette liste est mal configurée (vide ou aucune valeur correspondant au modèle de données), le module est uniquement accessible en consultation.

Pour déclarer un élément linéaire ou un élément nœud :

- ▶ **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné.
- ▶ **source** : type d'usage de l'équipement au niveau du réseau utility network  
Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction ».
- ▶ **assetGroup** : code de l'assetGroup correspondant à l'équipement au sein de la couche concernée.

La section « **filters** » permet de définir la liste des filtres de sélection personnalisés. Un filtre de sélection personnalisé permet d'affiner la portée de la sélection qui est faite dans la carte. Au lieu de sélectionner tous les équipements « sélectionnables », on ne va sélectionner que ceux qui répondent à la condition liée au filtre personnalisé actif.

Par exemple :

- ▶ Uniquement les équipements linéaires.
- ▶ Uniquement les tronçons qui sont dans le domaine « eau usée ».
- ▶ Uniquement les équipements qui sont visitables.
- ▶ Uniquement les tronçons qui sont en PVC, en acier ou en fonte et qui ont une longueur inférieure à 40m.

```
"filters": [
  {
    "name": "Eaux pluviales",
    "description": "Uniquement les équipements du domaine des eaux pluviales",
    "condition": {
      "whereClause": "reseau = 1"
    }
  },
  {
    "name": "Eaux usées",
    "description": "Uniquement les équipements du domaine des eaux usées",
    "condition": {
      "whereClause": "reseau = 2"
    }
  }
]
```

Figure 40 : Filtres attributaires pour les inspections du réseau

Des exemples de filtres personnalisés sont disponibles en [annexe](#).

Pour chaque filtre personnalisé :

- ▶ **name** : nom du filtre (obligatoire).
- ▶ **description** : description du filtre. Optionnel mais recommandé. Permet à l'utilisateur de savoir quelle est la portée du filtre et la(les) condition(s) appliquée(s).
- ▶ **condition** : condition à appliquer sur les équipements du réseaux lors de la sélection.
  - **networkEntities** : informations sur les types d'équipements qui peuvent être sélectionnés. Doit être un sous-ensemble du *networkEntities* déclaré au niveau supérieur. Optionnel.  
Pour chaque type :
    - **source** : type d'usage de l'équipement au niveau du réseau utility network.  
Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction » ;
    - **assetGroup** : code de l'AssetGroup correspondant à l'équipement au sein de la couche concernée ;

- **assetType** : code de l'AssetType correspondant au type d'équipement de l'AssetGrout concerné. Exemple : tronçon (= assetGroup) de type refoulement, décharge, simple, etc... (=assetType) ;
- **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné ;
- **whereClause** : condition qui s'applique soit sur les types d'équipements renseignés dans networkEntities (si networkEntities est renseigné) soit sur tous les types d'équipements sélectionnables pour le module (cfr. networkEntities du niveau supérieur).

#### 3.2.8.1.3 Configuration de la trace

La section « **traceConfigurations** » permet de définir la trace utilisée pour ordonner les éléments concernés et pour regrouper des tronçons lors de la création de l'inspection télévisée et du fichier canevas :

```
"traceConfigurations": [  
  {  
    "targetAssetGroups": [  
      200,  
      201  
    ],  
    "ignoreBarriersAtStartingPoints": false,  
    "barriers": {  
      "name": "Category",  
      "type": "category",  
      "operator": "equal",  
      "value": "ITV_ACCES",  
      "combineUsingOr": true,  
      "isSpecificValue": true  
    }  
  }  
],
```

*Figure 41 : Trace pour les inspections du réseau*

- **targetAssetGroups** : Liste des codes d'assetGroup cibles de la trace ;
- **ignoreBarriersAtStartingPoints** : Prise en compte de la barrière pour le point de départ.
- **barriers** : Définition de la configuration de barrière(s) pour la trace. Voir la documentation standard ESRI. La barrière est définie à l'aide d'une catégorie (ITV\_ACCES). Cette catégorie comprend la liste de tous les types de nœuds débouchant. Un nœud débouchant est un objet ponctuel du réseau par lequel on peut soit entrer soit sortir une caméra. Si l'administrateur veut modifier cette liste de nœuds débouchant alors il devra mettre à jour la catégorie ITV\_ACCES. S'il s'agit d'un nouveau type il faudra aussi mettre à jour la liste de nœuds dans la section networkEntities.

#### 3.2.8.1.4 Configuration des conséquences

La section « **consequences** » permet de configurer la remontée des conséquences à partir des données contenues dans le fichier ITV résultat.

Cette configuration est définie avec les balises suivantes :

- **tag** : correspond à un code, sur trois caractères, de la section ZB d'un fichier ITV.
- **csqType** : correspond au type de gestion de la conséquences avec les options suivantes :
  - **DIRECT** : la valeur de la conséquence, associée à la balise « tag », est directement recopiée ;

- **MAP** : la valeur de la conséquence, associée à la balise « tag », est convertie en s'appuyant sur la balise « lookupTable » décrite ci-dessous. La balise « tag » correspond à la clé pour récupérer la valeur.
- **COMPUTED** : pour les codes ACI, ACH et ACN, un calcul est effectué à partir de la valeur de l'« attribute2 » et du contenu du code pour déterminer la valeur de la conséquence.
- **mode** : correspond au mode de remplacement des conséquences avec les options suivantes :
  - **FIELD (0)** : la conséquence est systématiquement appliquée.
  - **BD (1)** : la conséquence est appliquée seulement si l'attribut n'est pas renseigné. Si l'attribut est renseigné et que sa valeur est différente de la valeur de la conséquence, une validation manuelle est nécessaire.
  - **MANUAL (2)** : la conséquence nécessite une validation manuelle sauf si la valeur de l'attribut est égale à celle de la conséquence.
- **domain** : domaine réseau des objets concernés.
- **layer** : numéro de la couche ou source, dans le domaine de réseau, des objets concernés.
- **assetGroup** : code de l'Asset Group, dans la source, des objets concernés.
- **attribute1** : nom de l'attribut associé au « tag » et à modifier.
- **attribute2** : nom d'un second attribut associé au « tag », pouvant être modifié, utilisé uniquement quand la balise « csqMode » vaut « COMPUTED » :
  - si tag vaut **ACH**, « attribute2 » désigne la profondeur du nœud d'arrivée selon le sens caméra ;
  - si tag vaut **ACI**, « attribute2 » désigne la profondeur du nœud de départ selon le sens caméra ;
  - si tag **ACN**, « attribute2 » désigne l'attribut la précision de la date de pose.
- **lookupTable** : liste de couples « clé et valeur » quand la balise « csqMode » vaut « MAP ».

```
{
  "tag": "ACB",
  "csqType": "DIRECT",
  "mode": "FIELD",
  "domain": "RES_ASS",
  "layer": 110,
  "assetGroup": 200,
  "attribute1": "hauteur"
},
{
  "tag": "ACA",
  "csqType": "MAP",
  "mode": "FIELD",
  "domain": "RES_ASS",
  "layer": 110,
  "assetGroup": 200,
  "attribute1": "forme",
  "lookupTable": {
    "A": "1",
    "B": "2",
    "C": "3",
    "D": "4",
    "E": "5",
    "F": "6",
    "X?": "",
    "Z": "99"
  }
},
{
  "tag": "ACH",
  "csqType": "COMPUTED",
  "mode": "FIELD",
  "domain": "RES_ASS",
  "layer": 110,
  "assetGroup": 200,
  "attribute1": "fromprf",
  "attribute2": "toprpf"
},
```

Figure 42 : Exemples de conséquence  
ITV

### 3.2.8.1.5 Configuration du plan d'inspection

La section « **canvasMap** », du fichier « **\config\onewater\sanitation\tvi.json** », permet de définir l'outil de génération du plan d'inspection télévisée utilisé lors de la génération du fichier canevas d'une

inspection préparée. Si cet outil est configuré, le fichier zip contient un fichier canevas au format XML et un plan.

! Cette fonctionnalité nécessite au minimum une version 11.3 d’ArcGIS Enterprise.

```
"canvasMap": {  
  "arcgisTool": {  
    "service": "default-service:1WATER/PrintItv",  
    "name": "PrintItv",  
    "task": "InspectionMap",  
    "tviGuidInputParameter": "programGuid",  
    "tokenInputParameter": "token",  
    "tokenRefererInputParameter": "tokenReferer",  
    "tokenExpiryInputParameter": "tokenExpiry",  
    "documentOutputParameter": "documentFile"  
  }  
}
```

*Figure 43 : Outil de génération du plan ITV*

Cette configuration est définie avec la balise suivante :

► **arcgisTool** : définition de l’outil.

L’outil est défini avec les balises suivantes :

- **service** : nom du service.
- **name** : nom de l’outil avec la valeur « PrintItv » (non modifiable).
- **task** : nom de la tâche avec la valeur « InspectionMap » (non modifiable).
- **tviGuidInputParameter** : nom du paramètre d’entrée donnant l’identifiant de l’inspection avec la valeur « programGuid » (non modifiable).
- **tokenInputParameter** : nom du paramètre d’entrée donnant le token d’identification au service avec la valeur « token » (non modifiable).
- **tokenRefererInputParameter** : nom du paramètre d’entrée donnant l’url référente du token avec la valeur « tokenReferer » (non modifiable).
- **tokenExpiryInputParameter** : nom du paramètre d’entrée donnant la durée de vie du token avec la valeur « tokenExpiry » (non modifiable).
- **documentOutputParameter** : nom du paramètre de sortie donnant l’url du plan généré avec la valeur « documentFile » (non modifiable).

Le fichier de configuration « **setting.json** », présent dans le répertoire du serveur ArcGIS (ex : D:\arcgisserver\1spatial\gpservices\printitv), permet de paramétrer le plan.

Il contient les paramètres suivants :

- **map** : informations sur la carte ;
  - **name** : le nom de la carte ;
  - **layers** : liste des couches nécessaires (doit contenir la couche **500**) ;
    - **id** : identifiant d’une couche ;
    - **name** : le nom de la couche ;
    - **attributes** : la liste des attributs visibles dans le document ;
  - **wkid** : numéro du système de référence ;
- **layout** : informations sur la mise en page ;
  - **name** : le nom de la mise en page ;
  - **éléments** : liste des éléments nécessaires (doit contenir les éléments Inspection, Localisation et Linéaire) ;
    - **name** : nom de l’élément ;



- **text** : modèle du texte de l'élément. {xxx} est remplacé par la valeur de l'attribut xxx listé dans le bloc **attributes** de la couche **500** ;
- **map** : informations sur la carte dans la mise en page ;
- **programZoomFactor** : coefficient de retrait appliqué à l'étendue de l'inspection pour définir le cadrage du plan.

Consulter le guide d'installation, au niveau du chapitre « Publier les services – Services de géotraitement – Publication du service d'impression du plan d'inspection » pour la mise en place du fichier « setting.json ».

### 3.2.8.2 Génération des fichiers de notations

L'exécutable **ParseScoreCCTVExcel.exe** livré avec la procédure d'installation du serveur est chargé de générer les fichiers de notations pour à la fois la méthode STANDARD et la méthode RERAU. L'outil s'appuie sur un fichier Excel qui décrit le paramétrage des notations. Un fichier standard est fourni avec l'outil et peut être modifié.

**ParseScoreCCTVExcel** est un exécutable de type Console. Les arguments de la ligne de commandes permettant de paramétrer la génération du fichier JSON sont les suivants :

Chaque argument est du type <Nom de l'argument>=<valeur de l'argument> :

- **FILE** : Chemin d'un fichier Excel (format .xlsx) qui contient la description des paramètres des méthodes STANDARD et RERAU. Le format de ce fichier Excel est décrit au paragraphe suivant) ;
- **ALPHA** : Valeur  $\alpha$  utilisé dans le calcul de la note élémentaire d'une observation ( $N=\alpha^P$ ). La valeur est un entier strictement positif, (2, 3 ou 4) ;
- **P** : Facteur multiplicateur à appliquer aux observations de type ponctuels dans le calcul de la note. P est un entier strictement positif (ex : 1) ;
- **TYPE** : Méthode de calcul des notes (« STANDARD » ou « RERAU ») selon que l'on souhaite générer le fichier de configuration de notation « STANDARD » ou « RERAU » ;
- **OUT** : Chemin du fichier JSON généré. Ce fichier sera au format JSON et généré avec un encodage UTF-8.

Dans le fichier Excel, les coefficients, les règles peuvent être modifiées selon le paramétrage souhaités. Cependant, un certain nombre de contraintes doivent être respectées.

Le fichier Excel contient 4 onglets qui doivent **impérativement** conserver le même nom et le même format.

A savoir :

- **1WATER\_SCORES** et **RERAU\_SCORES** contiennent la description du calcul pour chaque type d'observation. Ces onglets sont constitués de 5 colonnes fixes plus une colonne par indicateur de la note.
  1. Colonne 1 : désigne le code à 5 lettres de l'observation ;
  2. Colonne 2 : désigne le libellé du type d'observation ;
  3. Colonne 3 : cette colonne permet d'appliquer une règle supplémentaire sur l'observation et de distinguer ainsi plusieurs cas pour un même code d'observation. **ATTENTION**, pour un même indicateur, s'il existe plusieurs lignes avec un même code pour lequel l'indicateur est défini, toutes doivent avoir une règle ;

4. Colonne 4 : cette colonne permet de distinguer le type d'observation (A,E,I,R). Pour l'instant cette colonne n'est pas utilisée dans les calculs ;
  5. Colonne 5 : cette colonne permet de distinguer les observations ponctuelle (**P**) des observations linéaires (**L**) ;
  6. Colonne 6 à N : ces colonnes désignent le poids(**p**) à utiliser pour la ligne pour l'indicateur concerné. La note pour l'observation est définie par  $N = \alpha^P$ . Où **p** est le poids défini dans la colonne de l'indicateur. Si aucune valeur n'est définie, le type d'observation correspondant à la ligne n'est pas pris en compte dans le calcul de la note pour l'indicateur.
- **1WATER\_THRESHOLDS** et **RERAU\_THRESHOLDS** contiennent la description des seuils qui permettent de calculer les notes entières à partir des notes brutes. Chaque ligne est constituée de 4 colonnes fixes plus une colonne par intervalle de seuils. Les seuils sont donc définis par indicateurs et sont éventuellement différents pour des  **$\alpha$**  et des **P** distincts. **Le nom des indicateurs ne doit pas être changé :**
1. **Colonne 1 (A)** : définit le nom de l'indicateur pour lequel on définit le seuil ;
  2. **Colonne 2 (B)** : définit la valeur de  **$\alpha$**  pour lequel les seuils sont définis ;
  3. **Colonne 3 (C)** : définit le facteur multiplicateur à appliquer aux observations de type ponctuels dans le calcul de la note ;
  4. **Colonne 4 (D)** : définit le coefficient de pondération de l'indicateur dans le calcul de la note finale ;
  5. **Colonne 5 (E) à N** : Soit R la note brute du critère, et pour une colonne X soit v(X) la valeur de la cellule, alors :
    - Si  $0 \leq R < v(5)$  alors Note=1 ;
    - Si  $v(X-1) < R < v(X)$  pour  $5 < X < N$  alors Note = (X-4) ;
    - Si  $r \leq v(N)$  alors Note = N-3.

Les règles de la colonne 3 des onglets de description du calcul. 24 règles sont définies afin de filtrer les observations d'un même type (code à 5 lettres) :

1. **INF(v)** . Cette règle est réalisée si  $CODE\_D < v$  ;
2. **INFEQ(v)**. Cette règle est réalisée si  $CODE\_D \leq v$  ;
3. **SUP(v)** . Cette règle est réalisée si  $CODE\_D > v$  ;
4. **SUPEQ(v)**. Cette règle est réalisée si  $CODE\_D \geq v$  ;
5. **INTERV(inf, sup)**. Cette règle est réalisée si  $inf \leq CODE\_D < sup$  ;
6. **ECOUL(e)**. Cette règle porte sur le sens de l'écoulement dans l'équipement. Elle est réalisée si le sens d'écoulement e correspond au code AAK de l'inspection correspondante ;
7. **ENATURE( n)**. Cette règle porte sur la nature de l'équipement. Elle est réalisée si la nature n correspond au code ACK de l'inspection correspondante ;
8. **HPOS(min, max)**. Cette règle porte sur la position horaire de l'observation elle est réalisée si les deux codes [CODE\_G, CODE\_H] est inclus dans [min, max]. Si un des deux codes n'est pas rempli, il prend la valeur de l'autre code, définissant ainsi un intervalle de longueur 0 ;
9. **INF\_SMALL(v)**. Cette règle est réalisée si INF(v) est vérifié et que la longueur de l'observation est  $\leq 1$  ;
10. **INF\_LARGE(v)**. Cette règle est réalisée si INF(v) est vérifié et que la longueur de l'observation est  $> 1$  ;
11. **INFEQ\_SMALL(v)**. Cette règle est réalisée si INFEQ(v) est vérifié et que la longueur de l'observation est  $\leq 1$  ;

12. **INFEQ\_LARGE(v)**. Cette règle est réalisée si INFEQ(v) est vérifié et que la longueur de l'observation est > 1.
13. **SUP\_SMALL(v)**. Cette règle est réalisée si SUP(v) est vérifié et que la longueur de l'observation est <= 1 ;
14. **SUP\_LARGE(v)**. Cette règle est réalisée si SUP(v) est vérifié et que la longueur de l'observation est > 1 ;
15. **SUPEQ\_SMALL(v)**. Cette règle est réalisée si SUPEQ(v) est vérifié et que la longueur de l'observation est <= 1 ;
16. **SUPEQ\_LARGE(v)**. Cette règle est réalisée si SUPEQ(v) est vérifié et que la longueur de l'observation est > 1 ;
17. **INTERV\_SMALL(inf, sup)**. Cette règle est réalisée si INTERV(inf, sup) est vérifié et que la longueur de l'observation est <= 1 ;
18. **INTERV\_LARGE(inf, sup)**. Cette règle est réalisée si INTERV(inf, sup) est vérifié et que la longueur de l'observation est > 1 ;
19. **ENATURE\_INF(n, v)**. Cette règle est réalisée si ENATURE(n) et INF(v) sont vérifiés ;
20. **ENATURE\_SUP(n, v)**. Cette règle est réalisée si ENATURE(n) et SUP(v) sont vérifiés ;
21. **ENATURE\_INFEQ(n, v)**. Cette règle est réalisée si ENATURE(n) et INFEQ(v) sont vérifiés ;
22. **ENATURE\_SUPEQ(n, v)**. Cette règle est réalisée si ENATURE(n) et SUPEQ(v) sont vérifiés ;
23. **ENATURE\_IINTERV (n, inf, sup)**. Cette règle est réalisée si ENATURE(n) et INTERV(inf, sup) sont vérifiés ;
24. **ENATURE\_SUP\_HPOS (n, v1, inf1, sup1 [,v2, inf2, sup2])**. Cette règle est réalisée si ENATURE(n) et SUP(v1) et HPOS(inf1, sup1) sont vérifiés. Sinon si v2, inf2 et sup2 sont présents, la règle peut aussi être réalisée si ENATURE(n) et SUP(v2) et HPOS(inf2, sup2) sont vérifiés.

Après avoir généré le ou les fichiers JSON de configuration des notations il faut :

1. Renommer les fichiers.  
Pour la notation standard le fichier JSON doit se nommer  
« **tv\_i 1water score parameters.json** ».  
Pour la notation RERAU le fichier JSON doit se nommer « **tv\_i rerau score parameters.json** ».
2. Copier les fichiers dans le répertoire de configuration du serveur. Les deux fichiers doivent être placés dans le répertoire « **<config>/onewater/sanitation** » où <config> désigne le répertoire de configuration serveur.

### 3.2.8.3 Intégration des fichiers résultats

L'exécutable « **ITVVideoTranscoderServer.exe** », livré avec la procédure d'installation du serveur, est chargé d'observer un répertoire de dépôt. Quand un nouveau fichier résultat ITV (format ZIP) arrive dans ce répertoire, il est traité de la manière suivante :

- Les vidéos contenues dans le zip sont transcodées, quand cela est possible, dans un format utilisable par le web . Pour convertir les vidéos un outil open source standard (**ffmpeg.exe**) est utilisé. Cet outil permet de convertir une vidéo en un MPEG en utilisant le codec h264 nécessaire pour qu'un navigateur puisse lire la vidéo. La ligne de commande pour convertir la vidéo est la suivante :

```
ffmpeg.exe -i <video à convertir> -vcodec h264 -movflags faststart <video mpeg>
```

L'outil **ffmpeg.exe** inclus dans « **ITVVideoTranscoderServer.exe** » est en **version 4.4**.

- L'identifiant du programme ITV correspondant au fichier ITV est extrait :

- Si l'identifiant correspond à un programme ITV existant et valide (« Préparé » ou « En cours »), alors le fichier est déplacé sur le serveur 1Water dans le répertoire **<Répertoire permanent>/LINKED\_FILES** ;
- Si aucun programme d'ITV n'est associé à l'identifiant, alors le fichier est déplacé sur le serveur 1Water avec toute son arborescence dans le répertoire **<Répertoire permanent>/UNLINKED\_FILES**.

! Le traitement des fichiers déposés dans le répertoire est réalisé séquentiellement.

! La recherche de l'identifiant du programme d'inspection se fait en explorant l'attribut AAA des sections ZB. Si le fichier résultat correspond à un fichier « template » 1Water complété, une correspondance doit être trouvée.

L'exécutable « **ITVVideoTranscoderServer.exe** » est configuré avec les clés contenues dans le fichier de configuration « **ITVVideoTranscoderServer.exe.config** » :

- ▶ **DirIn** : Cette clé identifie le répertoire de dépôt des fichiers ITV à observer ;
- ▶ **Root1WaterUrl** : Cette clé définit la racine de l'URL d'accès au serveur 1Water. Elle permet à l'exécutable de se connecter au serveur ;
- ▶ **User** : Cette clé définit l'utilisateur de connexion au serveur 1Water. Il est possible d'encrypter l'utilisateur avec l'exécutable « **ITVVideoTranscoderServer.exe** ». L'utilisateur correspond à celui déclaré dans la ressource « **application.security.psa.user** » du fichier « **\config\onebiz.properties** » ;
- ▶ **Password** : Cette clé définit le mot de passe de l'utilisateur de connexion au serveur 1Water. Il est recommandé de crypter le mot de passe avec l'exécutable « **ITVVideoTranscoderServer.exe** ». Le mot de passe correspond à celui déclaré dans la ressource « **application.security.psa.password** » du fichier « **\config\onebiz.properties** » ;
- ▶ **SendTimeout** : En cas d'échec d'accès au serveur 1Water, l'opération sera retentée « au bout de » la valeur indiquée en millisecondes.

**Exemple de configuration :**

```
<appSettings>
  <add key="DirIn" value="D:\ITV\DEPOT" />
  <add key="Root1WaterUrl" value="http://<server>/1water-server" />
  <add key="User" value="<user>" />
  <add key="Password" value="<password>" />
  <add key="SendTimeout" value="5000" />
</appSettings>
```

! Modifier les clés en fonction de votre installation.

! L'encodage du fichier « **ITVVideoTranscoderServer.exe.config** » doit être en **UTF-8**.

L'exécutable « **ITVVideoTranscoderServer.exe** » peut être utilisé avec quatre modes différents :

- ▶ **Encryptage** : En mode console avec l'argument « **-encrypt** » suivi du mot de passe ou du nom d'utilisateur, retourne le mot de passe ou l'utilisateur encrypté ;
- ▶ **Console** : En mode console sans argument, il permet d'observer le répertoire « **DirIn** » et de traiter les fichiers résultats déposés ;
- ▶ **Service** : Sous la forme d'un service, avec l'argument « **-service** », il permet d'observer le répertoire « **DirIn** » et de traiter les fichiers résultats déposés.
- ▶ **Informations** : Ce mode est un mode console qui offre 2 options pour obtenir des informations de l'outil **ffmpeg.exe** :
  - « **-formats** » : permet de lister l'ensemble des formats de vidéo pris en charge par l'outil.

- « **-codecs** » : permet de lister l'ensemble des codecs pris en charge par l'outil.

Pour installer le service, effectuer les opérations suivantes :

- ▶ Exécuter la commande suivante dans une fenêtre de commandes :

```
sc create TviVideoTranscoderService binPath= "<répertoire>\ITVVideoTranscoderServer.exe -service"
```

! Remplacer « **répertoire** » par le répertoire d'installation de l'exécutable.

- ▶ Accéder aux services Windows et changer l'utilisateur du service « TviVideoTranscoder » au niveau de l'onglet « Connexion » et utiliser le compte « Service Réseau » ;
- ▶ Donner des droits en écriture, lecture, destruction de fichiers à l'utilisateur Service Réseau sur le répertoire « **Dirln** ».

## 3.2.9 Intervention

Le partie « Intervention », de l'application Assainissement permet de gérer différents modules selon la nature de l'intervention.

Chaque nature est associée à un fichier de configuration au format json.

1Water permet de gérer, dans cette version, les modules suivants :

- ▶ **Curage** : défini par le fichier de configuration « **\config\onewater\sanitation\cleaning.json** » ;
- ▶ **Visite** : défini par le fichier de configuration « **\config\onewater\sanitation\visit.json** ».

Chaque module est associé à des droits fonctionnels (Consultation, Edition, Gestion).

### 3.2.9.1 Configuration

La configuration d'un module d'intervention, pour une nature donnée, est définie par plusieurs sections dans le fichier json associé.

#### 3.2.9.1.1 Configuration générale

Cette section permet de définir les éléments nécessaires à la création, modification, destruction et gestion des interventions d'un module :

```
"buffer": 5.0,  
"maxItemsInList": 50,  
"canManageAnomaly" : true,
```

Figure 44 : Généralités pour intervention (Assainissement)

- ▶ **buffer** : distance en mètre utilisée pour construire la géométrie du programme d'intervention. Cette géométrie correspond au polygone créé à partir de la géométrie des canalisations qui constituent le programme à laquelle on applique une zone tampon de la taille indiquée ;
  - ▶ **maxItemsInList** : nombre maximum d'interventions qu'il est possible d'afficher dans la carte et dans la liste des interventions. Si dans l'étendue courante de la carte il y a plus d'interventions que la valeur de « maxItemsInList » :
    - L'affichage des éléments dans la carte bascule en mode « cluster » ;
    - La liste des éléments contient un message qui indique notamment le nombre d'interventions présentes (en lieu et place de la liste des interventions présentes dans la carte).
- ! Plus la valeur de ce paramètre est élevée, plus la fluidité d'affichage de la carte risque d'être dégradée.

- **canManageAnomaly** : Permet au module d'intervention d'autoriser ou non la gestion des anomalies.

### 3.2.9.1.2 Configuration du patrimoine

La section « **networkEntities** » permet de définir la liste des éléments du patrimoine concernés par le module d'intervention :

```
"networkEntities": [  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 200,  
    "domainNetworkName": "RES_EP"  
  },  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 202,  
    "domainNetworkName": "RES_EP"  
  },  
  {  
    "source": "esriUNFCUTDevice",  
    "assetGroup": 263,  
    "domainNetworkName": "RES_EP"  
  },  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 300,  
    "domainNetworkName": "RES_EU"  
  },  
  {  
    "source": "esriUNFCUTLine",  
    "assetGroup": 302,  
    "domainNetworkName": "RES_EU"  
  },  
  {  
    "source": "esriUNFCUTDevice",  
    "assetGroup": 363,  
    "domainNetworkName": "RES_EU"  
  }  
],
```

Figure 45 : Patrimoine pour les interventions (Assainissement)

Pour définir un type d'élément :

- **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné ;
- **source** : type d'usage de l'équipement au niveau du réseau utility network. Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction » ;
- **assetGroup** : code de l'assetGroup correspondant à l'équipement au sein de la couche concernée.

! Si cette liste est mal configurée (vide ou aucune valeur correspondant au modèle de données), le module est uniquement accessible en consultation.

La section « **filters** » permet de définir la liste des filtres de sélection personnalisés. Un filtre de sélection personnalisé permet d'affiner la portée de la sélection qui est faite dans la carte. Au lieu de sélectionner tous les équipements « sélectionnables », on ne va sélectionner que ceux qui répondent à la condition liée au filtre personnalisé actif.

Par exemple :

- Uniquement les équipements linéaires.
- Uniquement les tronçons qui sont dans le domaine « eau usée ».
- Uniquement les équipements qui sont visitables.
- Uniquement les tronçons qui sont en PVC, en acier ou en fonte et qui ont une longueur inférieure à 40m.

```
"filters": [
  {
    "name": "Eaux pluviales",
    "description": "Uniquement les équipements du domaine des eaux pluviales",
    "condition": {
      "whereClause": "reseau = 1"
    }
  },
  {
    "name": "Eaux usées",
    "description": "Uniquement les équipements du domaine des eaux usées",
    "condition": {
      "whereClause": "reseau = 2"
    }
  }
],
```

Figure 46 : Filtres attributaires pour les interventions (Assainissement)

Des exemples de filtres personnalisés sont disponibles en [annexe](#).

Pour chaque filtre personnalisé :

- ▶ **name** : nom du filtre (obligatoire).
- ▶ **description** : description du filtre. Optionnel mais recommandé. Permet à l'utilisateur de savoir quelle est la portée du filtre et la(les) condition(s) appliquée(s)
- ▶ **condition** : condition à appliquer sur les équipements du réseaux lors de la sélection
  - **networkEntities** : informations sur les types d'équipements qui peuvent être sélectionnés. Doit être un sous-ensemble du *networkEntities* déclaré au niveau supérieur. Optionnel.  
Pour chaque type :
    - **source** : type d'usage de l'équipement au niveau du réseau utility network. Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction » ;
    - **assetGroup** : code de l'assetGroup correspondant à l'équipement au sein de la couche concernée ;
    - **assetType** : code de l'assetType correspondant au type d'équipement de l'assetGroup concerné. Exemple : tronçon (= assetGroup) de type refoulement, décharge, simple, etc... (=assetType) ;
    - **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné ;
  - **whereClause** : condition qui s'applique soit sur les types d'équipements renseignés dans *networkEntities* (si *networkEntities* est renseigné) soit sur tous les types d'équipements sélectionnables pour le module (cfr. *networkEntities* du niveau supérieur).

La section « **linkedNodes** » permet de définir la liste des équipements de type nœud concernés par le graphe. Cette section définit la liste des nœuds, non concernés par les opérations d'intervention du module, mais pouvant se trouver à l'extrémité d'un tronçon ou d'un branchement concerné.

Par exemple, les points de collecte peuvent être situés à l'extrémité d'un branchement, mais ne sont pas concernés par les opérations de curage ou de visite. Les informations de cette section sont exploitées par la trace si la configuration ne s'appuie pas sur une trace nommée.



```
"linkedNodes": [
  {
    "domainNetworkName": "RES_EU",
    "source": "esriUNFCUTDevice",
    "assetGroup": 361
  },
  {
    "domainNetworkName": "RES_EU",
    "source": "esriUNFCUTJunction",
    "assetGroup": 324
  },
  {
    "domainNetworkName": "RES_EP",
    "source": "esriUNFCUTDevice",
    "assetGroup": 261
  },
  {
    "domainNetworkName": "RES_EP",
    "source": "esriUNFCUTJunction",
    "assetGroup": 224
  }
],
```

Figure 40 : Relations pour les interventions (Assainissement)

Pour définir un type d'élément :

- **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné ;
- **source** : type d'usage de l'équipement au niveau du réseau utility network. Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction » ;
- **assetGroup** : code de l'assetGroup correspondant à l'équipement au sein de la couche concernée.

### 3.2.9.1.3 Configuration de la trace

La section « **traceConfigurations** » permet de définir la trace utilisée pour ordonner les éléments concernés par l'intervention :

```
"traceConfigurations": [
  {
    "targetAssetGroups": [
      200,
      300
    ],
    "ignoreBarriersAtStartingPoints": false,
    "barriers": {
      "name": "Category",
      "type": "category",
      "operator": "equal",
      "value": "ITV_ACCES",
      "combineUsingOr": true,
      "isSpecificValue": true
    }
  }
]
```

Figure 47 : Trace pour les interventions (Assainissement)

- **targetAssetGroups** : Liste des codes d'Asset Group cibles de la trace ;
- **ignoreBarriersAtStartingPoints** : Prise en compte de la barrière pour le point de départ (Obsolète) ;
- **barriers** : Définition de la configuration de barrière(s) pour la trace (Obsolète). Voir la documentation standard ESRI.



## 3.2.10 Notation

### 3.2.10.1 Configuration du module

La configuration du module de notation est définie dans le fichier « `\config\onewater\sanitation\scoring.json` » :

```
"buffer": 5.0,
"networkEntities": [
  {
    "source": "esriUNFCUTLine",
    "assetGroup": 200,
    "domainNetworkName": "RES_ASS"
  },
  {
    "source": "esriUNFCUTLine",
    "assetGroup": 201,
    "domainNetworkName": "RES_ASS"
  }
],
"filters": [
  {
    "name": "Eaux pluviales",
    "description": "Uniquement les équipements du domaine des eaux pluviales",
    "condition": {
      "whereClause": "reseau = 1"
    }
  },
  {
    "name": "Eaux usées",
    "description": "Uniquement les équipements du domaine des eaux usées",
    "condition": {
      "whereClause": "reseau = 2"
    }
  }
],
"geographicSearchSources": [
  {
    "featureServiceUrl": "default:/1WATER_DEV/1DevSectorisation",
    "layerId": 1
  }
]
```

Figure 48 : Patrimoine pour la notation (Assainissement)

► **buffer** : taille en mètre de la zone tampon autour des ouvrages concernés par la notation ;

Pour définir un type d'élément (au niveau de la section « **networkEntities** ») :

- **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné ;
- **source** : type d'usage de l'équipement au niveau du réseau utility network. Valeurs possibles : « esriUNFCUTLine » ;
- **assetGroup** : code de l'assetGroup correspondant à l'équipement au sein de la couche concernée.

! Si cette liste est mal configurée (vide ou aucune valeur correspondant au modèle de données), le module est uniquement accessible en consultation.

Pour définir les objets utilisés comme source de la sélection géographique (au niveau de la section « **geographicSearchSources** ») :

- **featureServiceUrl** : Url vers le service de la couche ;
- **layerId** : Identifiant de la couche (au niveau du service).

La section « **filters** » permet de définir la liste des filtres de sélection personnalisés. Un filtre de sélection personnalisé permet d'affiner la portée de la sélection qui est faite dans la carte. Au lieu de sélectionner tous les équipements « sélectionnables », on ne va sélectionner que ceux qui répondent à la condition liée au filtre personnalisé actif. Par exemple :

- Uniquement les équipements linéaires

- ▶ Uniquement les équipements qui sont dans le domaine « eau usée »
- ▶ Uniquement les tronçons qui sont en PVC, en acier ou en fonte
- ▶ Etc...

Des exemples de filtres personnalisés sont disponibles en [annexe](#).

Pour chaque filtre personnalisé :

- ▶ **name** : nom du filtre (obligatoire).
- ▶ **description** : description du filtre. Optionnel mais recommandé. Permet à l'utilisateur de savoir quelle est la portée du filtre et la(les) condition(s) appliquée(s)
- ▶ **condition** : condition à appliquer sur les équipements du réseaux lors de la sélection
  - **networkEntities** : informations sur les types d'équipements qui peuvent être sélectionnés. Doit être un sous-ensemble du *networkEntities* déclaré au niveau supérieur. Optionnel.  
Pour chaque type :
- ▶ **source** : type d'usage de l'équipement au niveau du réseau utility network. Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction » ;
- ▶ **assetGroup** : code de l'assetGroup correspondant à l'équipement au sein de la couche concernée ;
- ▶ **assetType** : code de l'assetType correspondant au type d'équipement de l'assetGroup concerné. Exemple : tronçon (= assetGroup) de type refoulement, décharge, simple, etc... (=assetType) ;
- ▶ **domainNetworkName** : nom du domaine du réseau auquel est rattaché le type d'équipement concerné ;
- ▶ **whereClause** : condition qui s'applique soit sur les types d'équipements renseignés dans *networkEntities* (si *networkEntities* est renseigné) soit sur tous les types d'équipements sélectionnables pour le module (cfr. *networkEntities* du niveau supérieur).

### 3.2.10.2 Configuration des notes

La configuration des notes pour l'application est définie dans le fichier « `\config\onewater\ sanitation \score_definitions.json` » :

```
[
  {
    "name": "ALEA",
    "description": "Note d'aléa",
    "size": 4,
    "productName": "onewater",
    "moduleName": "sanitation",
    "formula": "STANDARD",
    "criteria": [
    ],
  },
  {
    "name": "ENJEU",
    "description": "Note d'enjeu",
    "size": 4,
    "productName": "onewater",
    "moduleName": "sanitation",
    "formula": "STANDARD",
    "criteria": [
    ],
  },
  {
    "name": "RISQUE",
    "description": "Note de risque",
    "size": 4,
    "productName": "onewater",
    "moduleName": "sanitation",
    "formula": "SUM",
    "criteria": [
    ],
  }
]
```

Figure 49 : Définition des notes (Assainissement)

---

Pour la description de ce fichier de configuration, vous référer au document « **1Water Méthode de notation** ».

### 3.2.11 Mise à jour

Le module « Mise à jour » permet de saisir, de mettre à jour et de supprimer les objets du patrimoine dans le cadre d'une session de travail privée et dédiée à l'utilisateur. Cette session de travail est ensuite validée et publiée pour être partagée avec les autres utilisateurs.

La configuration, qui permet de définir des prototypes d'enchaînement par domaine du réseau de distribution, est décrite dans le fichier « **\config\onewater\sanitation\update.json** ».

Exemple de prototype d'un point de collecte (eaux usées) :

```

"RES_ASS": {
  "261": {
    "ALL": [
      {
        "source": "esriUNFCUTDevice",
        "name": "Création d'un point de collecte (eaux usées)",
        "steps": [
          {
            "name": "info",
            "label": "Informations",
            "fields": [
              {
                "name": "ident",
                "type": "input",
                "defaultValue": "RGB_A-XXX",
                "label": "Numéro",
                "isRequired": true
              }
            ]
          },
          {
            "name": "draw",
            "label": "Dessiner",
            "message": "Dessiner un arc",
            "infos": [
              {
                "label": "1er point:",
                "value": "Position du regard de branchement"
              },
              {
                "label": "Dernier point:",
                "value": "Position du piquage"
              }
            ]
          }
        ]
      }
    ],
    "features": [
      {
        "source": "esriUNFCUTDevice",
        "assetGroup": 261,
        "assetType": 701,
        "fieldAttributes": "info",
        "attributes": {
          "accessible": 1,
          "reseau": 2
        }
      },
      {
        "source": "esriUNFCUTLine",
        "assetGroup": 204,
        "assetType": 701
      },
      {
        "source": "esriUNFCUTJunction",
        "assetGroup": 224,
        "assetType": 251,
        "reseau": 2
      }
    ]
  }
}

```

Figure 50 : Prototype « Point de collecte (eaux usées) »

**RES\_EU | RES\_EP | RES\_ASS** : Identifiant du domaine du réseau de distribution concerné par la liste des prototypes, avec pour chaque prototype :

- ▶ **#ID\_GROUP#**: Identifiant du groupe permettant de positionner la liste de prototypes dans l'arborescence de l'application ;
- ▶ **#ID\_TYPE#** ou **"ALL"**: Identifiant du type permettant de positionner la liste de prototypes dans l'arborescence de l'application ou ALL qui les placent au niveau chaque type;
- ▶ **source** : Identifiant de la source permettant de positionner le prototype dans l'arborescence de l'application ;
- ▶ **name** : Titre du panneau lors de l'ouverture du prototype ;
- ▶ **steps** : Liste des étapes du prototype (renseignement des attributs, saisie graphique des points) :
  - **name** : Nom de l'étape qui doit être unique ;
  - **label** : Libellé de l'étape qui apparaît au niveau de la progression du prototype ;
  - **message** : Message d'information affiché à l'utilisateur pour l'orienter dans l'action à effectuer (facultatif) ;

- 
- **fields** (renseignement des attributs) : Liste des champs à renseigner , pour le groupe associé au prototype courant, avant la création :
    - **name** : Nom du champ qui doit être unique et correspondre à un nom d'attribut de l'assetGroup ;
    - **type** : Type du champ ;
    - **assetTypes** : Liste déroulante des types disponibles pour le groupe du prototype ;
    - **input** : Champ texte éditable.
    - **label** : Libellé indiqué au niveau du champ ;
    - **defaultValue** : Valeur par défaut du champ ;
    - **isRequired** : Indique si le champ est obligatoire. Si le champ est obligatoire, l'étape ne pourra être validée sans renseigner le champ.
  - **infos** (saisie graphique de point) : Informations sur la marche à suivre pour l'étape courante :
    - **label** : Étape de saisie ;
    - **value** : Message associé à l'étape de saisie.
  - **features** : Liste des objets à créer. Chaque saisie ponctuelle, définie dans « **infos** » des étapes (**steps**), correspond à la création d'un objet ponctuel déclaré dans ce bloc et en respectant l'ordre. Les éléments linéaires sont obligatoirement déclarés entre deux saisies ponctuelles.
    - **source** : Identifiant de la source de l'objet à créer ;
    - **assetGroup** : Identifiant du groupe de l'objet à créer ;
    - **assetType** : Identifiant du type de l'objet à créer ;
    - **attributes** : Attributs à renseigner par défaut de l'objet :
      - **<clé>** : Nom de l'attribut ;
      - **<valeur>** : Valeur de l'attribut.
    - **fieldAttributes** : Nom (**name**) des étapes (**steps**) défini précédemment pour récupérer les couples « attribut / valeur » renseignées par l'utilisateur au niveau de « **fields** ».

Exemple de définition de prototypes différents par #ID\_TYPE# d'un même #ID\_GROUP# :

```
"RES_ASS": {
  "261": {
    "701": [
      {
        "source": "esriUNFCUTDevice",
        "name": "Création d'un point de collecte A",
        "steps": [
          "features": [
        ]
      },
      "702": [
        {
          "source": "esriUNFCUTDevice",
          "name": "Création d'un point de collecte B",
          "steps": [
            "features": [
          ]
        }
      ]
    ]
  }
}
```

Figure 20: Prototypes par type d'un même groupe (Assainissement)

### 3.2.12 Cadastre

Des fonctionnalités, liées au cadastre, sont proposées sous la forme de widget et permettent :

- ▶ de se localiser par référence cadastrale ;
- ▶ de se localiser par lieu-dit ;
- ▶ de consulter les informations d'une parcelle cadastrale.

Pour bénéficier de ces fonctionnalités, il est nécessaire d'avoir un accès à la base de données cadastre configuré dans le fichier « \config\cadastre.properties » :

```
cadastre.database.type=postgresql
cadastre.database.geometry.mode=pg_geometry
cadastre.database.geometry.srid=2154
cadastre.database.driver=org.postgresql.Driver
cadastre.database.url=jdbc:postgresql://lbz-d-lwags11.1spatial.fr:5432/dev_water
cadastre.database.user=cadastre
cadastre.database.password=cadastre
cadastre.database.initialSize=0
cadastre.database.maxTotal=8
cadastre.database.validationQuery=select version();
cadastre.database.schema.cadastre=cadastre
cadastre.database.schema.plu=cadastre
urba.directory=E:\\DATA\\Cadastre\\urba
```

Figure 51 : Configuration du cadastre sous PostgreSQL (Assainissement)

! Le fichier de configuration peut prendre deux formes selon que la base de données « Cadastre » soit sous Oracle ou PostgreSQL.

La signification de chaque clé est la suivante :

- ▶ **cadastre.database.type** : indique le type de base de données cadastre utilisé. Les valeurs possibles sont :
  - oracle
  - postgresql
- ▶ **cadastre.database.geometry.mode** : mode de géométrie pour la base de données. Les valeurs possibles sont :

- **st\_geometry** : géométrie d'ESRI ;
- **pg\_geometry** : si « `cadastre.database.type` » est égal à « `postgresql` » ;
- **sdo\_geometry** : si « `cadastre.database.type` » est égal à « `oracle` ».
- ▶ **cadastre.database.geometry.srid** : système de référence des géométries représentées dans la base de données.
- ▶ **cadastre.database.driver** : classe du driver de base de données. Les valeurs possibles sont :
  - **org.postgresql.Driver** : si « `cadastre.database.type` » est égal à « `postgresql` » ;
  - **oracle.jdbc.OracleDriver** : si « `cadastre.database.type` » est égal à « `oracle` ».
- ▶ **cadastre.database.url** : url de connexion à la base de données ; Selon le type de base de données la chaîne de connexion peut prendre différentes formes
  - si « `cadastre.database.type` » est égal à « `postgresql` », elle est du type **`jdbc:postgresql://<host>[<port>]/<nom de la base de données>`** avec :
    - **<host>** : nom de la machine sur laquelle est installée la base de données ;
    - **<port>** : numéro du port. Cette valeur est optionnelle (par défaut 5432).
  - si « `cadastre.database.type` » est égal à « `oracle` », elle est du type **`jdbc:oracle:thin:@//<host>[:<port>]/<service_name>`** avec :
    - **<host>** : nom de la machine sur laquelle est installée la base de données ;
    - **<port>** : numéro du port.
- ▶ **cadastre.database.user** : utilisateur de connexion à la base de données.
- ▶ **cadastre.database.password** : mot de passe de l'utilisateur de connexion. Le mot de passe de l'utilisateur de connexion peut être crypté.  
Le fichier « **Encrypt.bat** », présent au niveau du répertoire racine du serveur 1BiZ, permet d'encrypter un mot de passe.

```
C:\1BiZServer>Encrypt.bat abcd
cryptage...
abcd: <xMIIRx5J19InuoTO+0Pr2zx8N2YE7L2QPz1T+k03+Vo=>
```

- ! Copier le contenu entre <> au niveau de la clé dans le fichier de configuration et le faire précéder d'un ~ (tilde).

Exemple : `cadastre.database.password=~xMIIRx5J19InuoTO+0Pr2zx8N2YE7L2QPz1T+k03+Vo=`

- ▶ **cadastre.database.initialSize** : taille initiale du pool de connexion pour les connexions à la base de données.
- ▶ **cadastre.database.maxTotal** : taille maximale du pool de connexion pour les connexions à la base de données.
- ▶ **cadastre.database.validationQuery** : requête utilisée pour valider la connexion à la base de données. En général, on utilise :
  - **`select version();`** : si « `cadastre.database.type` » est égal à « `postgresql` » ;
  - **`select 1 from dual;`** : si « `cadastre.database.type` » est égal à « `oracle` ».
- ▶ **cadastre.database.schema.cadastre** : nom du schéma associé au cadastre dans la base de données.
- ▶ **cadastre.database.schema.plu** : nom du schéma contenant le plan d'occupation des sols (PLU).
- ▶ **urba.directory** : chemin vers le répertoire contenant les documents d'urbanisme.

! Les clés, dans le fichier de configuration livré, ne sont pas renseignées par défaut. Elles doivent toutes être configurées pour accéder aux fonctionnalités du cadastre.

### 3.2.13 Gestion des conséquences

La fonctionnalité « Gestion des conséquences » n'est pas à proprement dit un module, mais une fonctionnalité « transverse ». Une conséquence correspond à la modification d'un attribut sur un élément du patrimoine.

Les conséquences peuvent provenir :

- d'une information remontée par la mobilité (ex. : lors de la réparation d'une anomalie) ;
- d'un fichier résultat d'une inspection télévisée.

La configuration de la fonctionnalité « Gestion des conséquences » est définie dans le fichier « **\config\onewater\sanitation\consequences.json** » : par défaut, ce fichier contient une liste vide ({}).

Le format est le suivant :

```
{
  "networkEntities": [
    {
      "domainNetworkName": "<Nom du domaine de réseau>",
      "source": "<Nom de la source UN>",
      "assetGroup": "<Code de l'AssetGroup>",
      "attributes": [
        {
          "name": "<Nom de l'attribut>",
          "replaceMode": "<Mode de remplacement de la valeur de l'attribut>"
        }
      ]
    }
  ]
}
```

Les paramètres sont :

- **networkEntities** : la liste des types d'équipements qui peuvent faire l'objet d'une conséquence ;
  - **domainNetworkName** : Texte, obligatoire, pour le domaine de réseau;
  - **source** : Texte, obligatoire, pour la source;
  - **assetGroup** : Entier, obligatoire, pour le code de l'AssetGroup ;
  - **attributes** : la liste des attributs, pouvant faire l'objet d'une conséquence, pour l'AssetGroup ;
    - **name** : Texte (max 50 caractères), obligatoire, pour le nom de l'attribut ;
    - **replaceMode** : Entier, facultatif, pour le mode de remplacement lors de la gestion des conséquences sur le serveur :
      - 0 : la conséquence est systématiquement appliquée (valeur par défaut) ;
      - 1 : la conséquence est appliquée seulement si l'attribut n'est pas renseigné. Si l'attribut est renseigné et que sa valeur est différente de la valeur de la conséquence, une validation manuelle est nécessaire ;
      - 2 : la conséquence nécessite une validation manuelle sauf si la valeur de l'attribut est égale à celle de la conséquence.



Exemple de fichier de configuration de gestion des conséquences :

```
{
  "networkEntities": [
    {
      "assetGroup": 201,
      "domainNetworkName": "RES_ASS",
      "source": "esriUNFCUTLine",
      "attributes": [
        {
          "name": "adresse",
          "replaceMode": 0
        },
        {
          "name": "insee",
          "replaceMode": 0
        }
      ]
    }
  ]
}
```

## 4 CONFIGURATION DE LA MOBILITÉ

### 4.1 AUTHENTIFICATION OAUTH/SSO

Dans le cadre d'une authentification OAuth/SSO, il est impératif que l'application 1Water Mobile soit enregistrée sur le portail ESRI permettant à l'utilisateur de s'authentifier.

! Une connexion SSO est fortement déconseillée si une tablette est partagée entre plusieurs utilisateurs car la déconnexion n'est pas effective.

Pour ajouter et inscrire une application, suivre la [documentation ESRI](#).

Exemple de paramètres pour ajouter et inscrire l'application mobile :

- Sélectionnez le type d'application : choisir obligatoirement **Mobile** ;
- Pour Web Mapping (Cartographie web), Desktop (Bureau) ou Mobile (Mobile) : saisissez l'URL de l'application (ex : <https://1spatial.com/fr/>) ;
- Spécifiez un titre (ex : 1Water Mobile) ;
- Dans My content (Mon contenu), choisissez un dossier dans lequel l'élément doit être enregistré : sélectionner le dossier des applications 1Water (ex : 1Water).

Après l'ajout, vous devez paramétrer et enregistrer l'application 1Water Mobile.

! Cet enregistrement est indispensable pour permettre de renseigner la valeur de l'attribut `appID` dans le fichier de configuration de l'application 1Water Mobile (cf. 1Water Guide Installation au niveau du chapitre Fichier « applicationConfig.json »).

Sur la page « Paramètres » de l'application 1Water Mobile :

- Recommander l'utilisation de cet élément : cliquer sur le bouton « Marquer comme officiel » pour activer la protection ;
- Enregistrement de l'application (bas de page) : cliquer sur le bouton « Enregistrer ».
  - Le panneau d'enregistrement s'ouvre (son aspect peut varier suivant la version d'ArcGIS) :

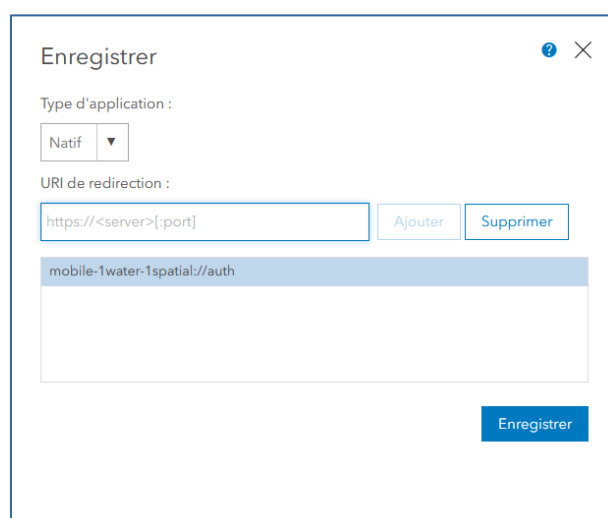


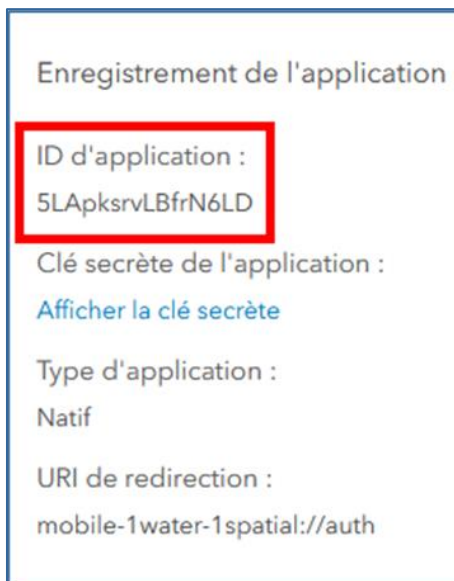
Figure 52 : Panneau d'enregistrement d'une application

- Type d'application : sélectionner **Natif** ;
  - o Ajouter l'URI de redirection suivante : **mobile-1water-1spatial://auth**

! Ne pas modifier cette URI car cela empêchera la connexion de l'application 1Water Mobile au portail. Vérifier, en cas de Copier/Coller, qu'aucun espace n'a été ajouté.

- o Cliquer sur le bouton « Enregistrer » pour terminer l'enregistrement.

Une application correctement enregistrée présente cet aspect :



Enregistrement de l'application

ID d'application :  
5LApksrvLBfrN6LD

Clé secrète de l'application :  
[Afficher la clé secrète](#)

Type d'application :  
Natif

URI de redirection :  
mobile-1water-1spatial://auth

Figure 53 : Application enregistrée avec son identifiant

! La valeur associée à « ID d'application » doit être recopier dans l'attribut `appID` du fichier de configuration de l'application 1Water Mobile

## 4.2 FICHIERS DE CONFIGURATION SUR LE MOBILE

Avant de copier le répertoire « *fr.onespatial.onewater* » sur le mobile, il faut mettre à jour les fichiers de configuration du répertoire « *config* ».

### 4.2.1 Fichier « `applicationConfig.json` »

Ce fichier permet de définir les paramètres principaux de l'application :

```
{
  "network": {
    "portalUrl": "$PORTAL_URL$",
    "serverUrl": "$SERVER_URL$",
    "appID": "$APP_ID$"
  },
  "backgroundFile": "$BACKGROUND_FILE$",
  "spatialReference": "$SPATIAL_REFERENCE_ID$ sans guillemet",
```

```
"userAreaSpatialReference" : "$SPATIAL_REFERENCE_ID$ sans guillemet",  
"realPrecision": 0.001,  
"realDecimalDisplay": 3,  
"downloadTimeout": 600,  
"synchronizeTimeout": 120  
}
```

- ▶ **network** : Ensemble des paramètres de connexion.
    - **portalUrl** : URL d'accès au portail ESRI permettant à l'utilisateur de s'authentifier. Remplacer la chaîne « \$PORTAL\_URL\$ » par votre URL.
    - **serverUrl** : URL d'accès au serveur 1BiZServer, qui retourne les informations de configuration de l'application. Remplacer la chaîne « \$SERVER\_URL\$ » par votre URL ;
    - **appId** : ID de l'application dans le cadre d'une authentification OAuth/SSO (Optionnel). Remplacer la chaîne « \$APP\_ID\$ » par l'ID de l'application 1Water Mobile enregistrée sur le portail ESRI permettant à l'utilisateur de s'authentifier.
  - ▶ **backgroundFile** : Nom du fichier TPK ou VTPK utilisé pour afficher le fond de plan dans la carte. Ce fichier doit être placé dans le répertoire « data/background\_image ». Remplacer la chaîne « \$BACKGROUND\_FILE\$ » par votre nom de fichier ;
  - ▶ **spatialReference** : Identifiant ESRI du système de coordonnées (WKID précédent) utilisé pour l'affichage des données géographiques. Remplacer la chaîne « \$SPATIAL\_REFERENCE\_ID\$ » par votre identifiant. Ce paramètre est une valeur numérique, ne pas mettre de guillemet ;
  - ▶ **userAreaSpatialReference** : Identifiant ESRI du système de coordonnées utilisé pour la géométrie de la zone de travail (restriction) de l'utilisateur. Remplacer la chaîne « \$SPATIAL\_REFERENCE\_ID\$ » par votre identifiant. Ce paramètre est une valeur numérique, ne pas mettre de guillemet ;
  - ▶ **realPrecision** : Précision de comparaison des valeurs réelles (par défaut 0.001) ;
  - ▶ **realDecimalDisplay** : Nombre de décimales affichées pour les valeurs réelles (par défaut 3) ;
  - ▶ **downloadTimeout** : Temps maximum pour le téléchargement d'un réplica en secondes, (par défaut 600 secondes) ;
  - ▶ **synchronizeTimeout** : Temps maximum pour la synchronisation d'un réplica en secondes, (par défaut 120 secondes).
- ! Saisir l'URL du portail dans un navigateur et vérifier que la page de connexion s'affiche normalement. En cas de problème lié à un certificat de sécurité, corriger celui-ci car, contrairement au navigateur, 1Water ne peut pas contourner cette erreur.



#### Votre connexion n'est pas privée

Des individus malveillants tentent peut-être de subtiliser vos informations personnelles sur le site **1Spatial** (mots de passe, messages ou numéros de carte de crédit, par exemple). [En savoir plus](#)

NET::ERR\_CERT\_AUTHORITY\_INVALID

💡 Pour bénéficier du niveau de sécurité le plus élevé de Chrome, [activez la protection renforcée](#)

Paramètres avancés

Revenir en lieu sûr

Figure 54 : Problème de certificat de sécurité

## 4.2.2 Fichier « logConfig.json »

Ce fichier permet de définir le niveau des messages de log de l'application.

```
{  
  "logLevel": "ERROR",  
  "saveJsonToFile" : false  
}
```

- **logLevel** : Niveau d'affichage ou d'écriture des logs. Les valeurs possibles, dans l'ordre décroissant du niveau de détail, sont VERBOSE, DEBUG, INFO, WARN, ERROR et FATAL. La valeur par défaut est ERROR.

! L'utilisation des niveaux VERBOSE, DEBUG ou INFO peut affecter les performances de l'application, leur usage doit être réservé pour l'analyse de cas de dysfonctionnement.

- **saveJsonToFile** : Option d'enregistrement des fichiers de configuration retournés par le serveur au format JSON dans le dossier « *Android/data/fr.onespatial.onewater/files/debug* ». Un sous-dossier portant la date et l'heure du démarrage de l'application est automatiquement créé.

Cet enregistrement permet de transmettre les fichiers de configuration au support 1Spatial pour analyse en cas de dysfonctionnement.

Les valeurs admises sont « true » (enregistrement activé) ou « false » (enregistrement désactivé).

Les fichiers ne sont créés que si le niveau de log donné par le paramètre « logLevel » est positionné sur « VERBOSE » ou « DEBUG ». La valeur par défaut de ce paramètre est « false ».

## 4.2.3 Fichier « userConfig.json »

Ce fichier contient les paramètres utilisateur permettant de modifier le comportement ou l'aspect graphique de certains éléments de l'application.

```
{  
  "scaleBar": {  
    "unit": 0,  
    "displayTime": 10,  
    "color": "#000000",  
    "textSize": 15  
  }  
}
```

- **ScaleBar** : Ensemble des paramètres concernant la barre d'échelle :

- **Unit** : Unités utilisées dans la barre d'échelle. Les valeurs admises sont :
  - 0 : Mètre (par défaut) ;
  - 1 : Pied ;
  - 2 : Mètre + Pied.
- **DisplayTime** : Temps en seconde d'affichage de la barre d'échelle. Les valeurs admises sont :
  - -1 : toujours affichée
  - 0 : jamais affichée ;
  - > 0 : temps d'affichage avant disparition (par défaut 10 secondes).
- **Color** : Couleur de la barre d'échelle au format hexadécimal « #AARRGGBB » avec sont :
  - <AA> = transparence ;
  - <RR> = composante rouge ;
  - <GG> = composante vert ;
  - <BB> = composante bleu.

La valeur par défaut est fixée à « #00000000 ».

**TextSize** : Taille en density-independent pixels (dp) du texte. (par défaut 15).

## 4.2.4 Gestion de la sécurité

Le fichier « *1Spatial.bks* » situé dans le répertoire « *security* » permet de valider les certificats SSL auto-signés des serveurs utilisés. Ce fichier doit éventuellement être personnalisé avec un outil approprié pour y ajouter les certificats nécessaires. Le fichier livré permet d'accéder aux serveurs 1Spatial.

## 4.3 FICHIERS DE CONFIGURATION SUR LE SERVEUR

Les fichiers de configuration sont présents, au niveau de 1BiZ Serveur, dans le répertoire « `server\apache-tomcat\webapps\onebiz-server\WEB-INF\classes\appparams` ».

Ce répertoire contient les sous-répertoires suivants :

- **1Biz** : contient les fichiers de configuration générale du serveur 1BiZ ;
- **1water-water** : contient les fichiers de configuration de l'application « Eau » ;
- **1water-sanitation** : contient les fichiers de configuration de l'application « Assainissement » ;

## 4.4 GESTION DES FILTRES

Les fichiers de filtres, déclarés dans les [répertoires de configuration](#), sont utilisés au niveau de 1Water pour :

- définir des conditions d'affichage par défaut d'une couche métier (Anomalie, Encrassement) ;
- définir des conditions de mise à jour, suppressions d'une couche métier ;
- définir des contraintes d'édition sur les attributs ;
- définir des conditions sur les attributs obligatoires.

Un filtre se présente sous la forme d'un fichier JSON contenant soit :

- une condition simple ;
- une combinaison de conditions simples appelée « filtre simple » ;
- une combinaison entre une condition simple et un filtre simple, ou entre 2 filtres simples (ou complexes) appelée « filtre complexe ».

## 4.5 CONDITION SIMPLE

Une condition simple se décrit de la manière suivante :

```
{  
  "type": "fr.onespatial.filter.SimpleCondition",  
  "mAttribute": "<Attribut utilisé par le filtre>",  
  "mTypeValue": "<Type de l'attribut>",  
  "mOperator": "<Opérateur de comparaison>",  
  "mFlags": <Option de comparaison>,  
  "mDateDelay": <Délai à appliquer dans la comparaison des dates>,  
  "mStringValue" ou "mIntValue" ou "mDoubleValue" ou "mDateValue" ou "mStringList"  
  | "mIntValue" ou "mSecondAttribute" ou "mGetValueMethodName": <Valeur utilisée  
pour la comparaison>  
}
```

### 4.5.1 Champs obligatoires

Les champs obligatoires d'une condition simple sont :

- ▶ **type** : Type de condition utilisé pour le filtre. Pour une condition simple, le type prend la valeur `fr.onespatial.filter.SimpleCondition`. La valeur est à mettre entre guillemets.
- ▶ **mAttribute** : Nom de l'attribut, pas son alias, sur lequel la condition du filtre s'appuie. La valeur est à mettre entre guillemets.
- ▶ **mTypeValue** : Type de l'attribut utilisé pour le filtre. La valeur est à mettre entre guillemets.  
Les types possibles sont :
  - `STRING_TYPE` : type chaîne de caractères ;
  - `INT_TYPE` : type entier ;
  - `DOUBLE_TYPE` : type double ;
  - `DATE_TYPE` : type date ;
  - `STRING_LIST_TYPE` : type liste de chaînes de caractères ;
  - `INT_LIST_TYPE` : type liste d'entiers.
- ▶ **mOperator** : Opérateur de comparaison utilisé pour le filtre. La valeur est à mettre entre guillemets.  
Les opérateurs de comparaison disponibles sont :
  - `EQUAL` : Egale à ;
  - `NOT_EQUAL` : N'est pas égal à ;
  - `LOWER` : Strictement inférieur à ;
  - `LOWER_OR_EQUAL` : Inférieur ou égal ;
  - `GREATER` : Strictement supérieur à ;
  - `GREATER_OR_EQUAL` : Supérieur ou égal ;
  - `START_WITH` : Commence par ;
  - `DO_NOT_START_WITH` : Ne commence pas par ;
  - `END_WITH` : Termine par ;
  - `DO_NOT_END_WITH` : Ne se termine pas par ;

- CONTAIN : Contient ;
- DO\_NOT\_CONTAIN : Ne contient pas ;
- IN : Contenu dans cette liste de valeur ;
- NOT\_IN : N'est pas contenu dans cette liste de valeur.

Les opérateurs de comparaison possibles dépendent du [type d'attribut](#) :

- chaîne de caractères (STRING\_TYPE) : EQUAL, NOT\_EQUAL, CONTAIN, DO\_NOT\_CONTAIN, START\_WITH, DO\_NOT\_START\_WITH, END\_WITH, DO\_NOT\_END\_WITH, IN, NOT\_IN ;
- entier ou chaîne de caractères contenant des valeurs entières (INT\_TYPE) : EQUAL, NOT\_EQUAL, LOWER, LOWER\_OR\_EQUAL, GREATER, GREATER\_OR\_EQUAL, IN, NOT\_IN ;
- double ou chaîne de caractères contenant des valeurs décimales (DOUBLE\_TYPE) : EQUAL, NOT\_EQUAL, LOWER, LOWER\_OR\_EQUAL, GREATER, GREATER\_OR\_EQUAL ;
- date (DATE\_TYPE) : EQUAL, NOT\_EQUAL, LOWER, LOWER\_OR\_EQUAL, GREATER, GREATER\_OR\_EQUAL ;
- liste de chaînes (STRING\_LIST\_TYPE) : IN, NOT\_IN ;
- liste de d'entiers (INT\_LIST\_TYPE) : IN, NOT\_IN.

► **<value>** : Valeur de comparaison pouvant s'exprimer de différentes façons :

- **mSecondAttribute** : Nom de l'attribut, entre guillemets, de comparaison à lire pour le filtre ou mot clé [@@today@@](#).
- **mGetValueMethodName** : Nom de la méthode donnant en retour la valeur à utiliser pour le filtre.

Indiquer une valeur spécifique. Le nom du champ varie en fonction du type de comparaison (mTypeValue) :

- mStringValue : Nom du champ pour une valeur de type « STRING\_TYPE » ;
- mIntValue : Nom du champ pour une valeur de type « INT\_TYPE » ;
- mDoubleValue : Nom du champ pour une valeur de type « DOUBLE\_TYPE » ;
- mStringList : Nom du champ pour une valeur de type « STRING\_LIST\_TYPE » ;
- mIntList : Nom du champ pour une valeur de type « INT\_LIST\_TYPE ».

! Le type d'attribut « DATE\_TYPE » est géré uniquement avec « mSecondAttribute » ou « mGetValueMethodName ».

! La valeur attribuée à ce champ doit être conforme au format JSON avec des guillemets pour les chaînes de caractères, l'utilisation des crochets pour déclarer une liste et pas de guillemet pour les valeurs de type entier ou double.

! Si mSecondAttribute est vide, le contenu de mGetValueMethodName est examiné puis, si le contenu de mGetValueMethodName est vide, c'est la valeur décrite dans le champ typé qui est examinée.

## 4.5.2 Champs optionnels

Les champs optionnels d'une condition simple sont :

- **mFlags** : Option de comparaison. La valeur est à mettre entre guillemets.  
Les options possibles sont :



- **STRING\_CASE\_INSENSITIVE** : permet de définir une condition non sensible à la casse pour les types chaîne de caractères. Par défaut, la condition est sensible à la casse.
  - **DATE\_USE\_DATE** : permet de définir une condition tenant compte uniquement de la date pour les attributs de type date. Par défaut, c'est une stricte égalité. Il est donc recommandé de préciser cette option pour les attributs de type date.
  - **DATE\_USE\_TIME** : permet de définir une condition tenant compte de la date et de l'heure pour les attributs de type date. Par défaut, c'est une stricte égalité. Il est donc recommandé de préciser cette option pour les attributs de type date.
- **mDateDelay** : Délai à appliquer dans la comparaison des dates. Le format utilisé est de la forme « PnYnMnD » avec un signe optionnel à préciser devant « P », « nY », « nM » et « nD ». La valeur est à mettre entre guillemets. Pour plus de précisions, référez-vous à la documentation de l'API de la plate-forme Android pour la [gestion des périodes](#).

### 4.5.3 Mots-clés

Certains mots-clés peuvent être utilisés à la place des noms d'attribut :

- **@@\_today\_@@** : Nom du pseudo attribut représentant la date du jour et permettant de définir un filtre sur les dates avec cette valeur ; Il est utilisé uniquement pour les attributs date avec le champ mSecondAttribute.

### 4.5.4 Méthodes disponibles

Les méthodes décrites ci-dessous sont utilisables avec le champ mGetValueMethodName :

- "fun fr.onespatial.preferences.UserPreferences.Companion.getCurrentUser(): String" : retourne le nom de l'utilisateur courant ;
- "fun fr.onespatial.preferences.UserPreferences.Companion.getLastUpload(): java.util.GregorianCalendar" : Retourne la date de la dernière synchronisation.

### 4.5.5 Exemples

Condition simple sur l'attribut « etat » de type entier qui doit être égal à 1.

```
{  
  "type": "fr.onespatial.filter.SimpleCondition",  
  "mAttribute": "etat",  
  "mOperator": "EQUAL",  
  "mIntValue": 1,  
  "mTypeValue": "INT_TYPE"  
}
```

Condition simple sur l'attribut « date\_cons » de type date, avec une précision à l'heure, qui doit être strictement supérieur à la date et heure de la dernière synchronisation.

```
{  
  "type": "fr.onespatial.filter.SimpleCondition",  
  "mAttribute": "date_cons",  
  "mOperator": "GREATER",  
  "mGetValueMethodName": "fun  
fr.onespatial.preferences.UserPreferences.Companion.getLastUpload():  
java.util.GregorianCalendar",  
}
```

```
"mTypeValue": "DATE_TYPE",  
"mFlags": "DATE_USE_TIME"  
}
```

## 4.6 FILTRE SIMPLE

Un filtre simple est un filtre qui contient plusieurs conditions simples combinées avec un seul et unique opérateur.

```
{  
  "type": "fr.onespatial.filter.SimpleFilter",  
  "mCombination": "<Opérateur de combinaison>",  
  "mConditions": [  
    {  
      Condition simple 1  
    },  
    {  
      Condition simple 2  
    }  
  ]  
}
```

### 4.6.1 Champs obligatoires

Dans un filtre simple, les champs obligatoires sont :

- **type** : Type de filtre. Pour un filtre simple, le type prend la valeur `fr.onespatial.filter.SimpleFilter`. La valeur est à mettre entre guillemets.
- **mCombination** : Opérateur de combinaison des conditions simples. Les opérateurs possibles sont « AND » ou « OR ». La valeur est à mettre entre guillemets.
- **mConditions** : Tableau, au format JSON, de [conditions simples](#) définissant le filtre et combinées avec l'opérateur défini dans « mCombination ».

### 4.6.2 Exemple

Filtre simple avec l'attribut « creationdate » qui doit être strictement inférieur à la date du jour en prenant en compte uniquement la date OU strictement inférieur à la date de dernière synchronisation en prenant en compte la date et l'heure.

```
{  
  "type": "fr.onespatial.filter.SimpleFilter",  
  "mCombination": "OR",  
  "mConditions": [  
    {  
      "mAttribute": "creationdate",  
      "mOperator": "LOWER",  
      "mTypeValue": "DATE_TYPE",  
      "mFlags": "DATE_USE_DATE"  
    },  
    {  
      "mAttribute": "creationdate",  
      "mOperator": "LOWER",  
      "mGetValueMethodName": "fun"
```

```
fr.onespatial.preferences.UserPreferences.Companion.getLastUpload():  
java.util.GregorianCalendar",  
    "mTypeValue": "DATE_TYPE",  
    "mFlags": "DATE_USE_TIME"  
}  
]  
}
```

## 4.7 FILTRE COMPLEXE

Un filtre complexe est une combinaison de [conditions simples](#) et/ou de [filtres simples](#) et/ou de filtres complexes. Ce type de filtre doit être utilisé lorsque les conditions simples sont combinées avec au moins deux opérateurs différents.

```
{  
  "type": "fr.onespatial.filter.ComplexFilter",  
  "mFirstFilter": {  
    Condition simple 1 ou Filtre simple 1 ou Filtre complexe 1  
  },  
  "mCombinationOper": "AND",  
  "mSecondFilter": {  
    Condition simple 2 ou Filtre simple 2 ou Filtre complexe 2  
  }  
}
```

### 4.7.1 Champs obligatoires

Dans un filtre complexe, les champs obligatoires sont :

- **type** : Type de filtre. Pour un filtre complexe, le type prend la valeur `fr.onespatial.filter.ComplexFilter`. La valeur est à mettre entre guillemets.
- **mFirstFilter** : Condition simple ou Filtre simple ou Filtre complexe.
- **mSecondFilter** : Condition simple ou Filtre simple ou Filtre complexe.
- **mCombinationOper** : Opérateur de combinaison des filtres « mFirstFilter » et « mSecondFilter ». Les opérateurs possibles sont « AND » ou « OR ». La valeur est à mettre entre guillemets.

### 4.7.2 Exemple

Filtre complexe basé sur une condition simple sur l'attribut « etat » de type entier qui doit être égal à 1 ET un filtre simple avec l'attribut « creationdate » qui doit être strictement inférieur à la date du jour en prenant en compte uniquement la date OU strictement inférieur à la date de dernière synchronisation en prenant en compte la date et l'heure.

```
{  
  "type": "fr.onespatial.filter.ComplexFilter",  
  "mFirstFilter": {  
    "type": "fr.onespatial.filter.SimpleCondition",  
    "mAttribute": "etat",  
    "mOperator": "EQUAL",  
    "mIntValue": 1,  
    "mTypeValue": "INT_TYPE"  
  },  
  "mCombinationOper": "AND",  
  "mSecondFilter": {
```

```
"type": "fr.onespatial.filter.SimpleFilter",
"mCombination": "OR",
"mConditions": [
  {
    "mAttribute": "creationdate",
    "mOperator": "LOWER",
    "mTypeValue": "DATE_TYPE",
    "mFlags": "DATE_USE_DATE"
  },
  {
    "mAttribute": "creationdate",
    "mOperator": "LOWER",
    "mGetValueMethodName": "fun
fr.onespatial.preferences.UserPreferences.Companion.getLastUpload():
java.util.GregorianCalendar",
    "mTypeValue": "DATE_TYPE",
    "mFlags": "DATE_USE_TIME"
  }
]
}
```

## 4.8 CONFIGURATION DE LA RECHERCHE PAR ADRESSE

La recherche « Recherche par adresse » est un type optionnel de recherche qui apparaîtra dans le module « Recherche par adresse et par identifiant ». Ce type de recherche dépend de la mise en place d'un service de géocodage (Locator) dans ArcGIS Pro et de la disponibilité d'un fichier locator.zip.

Pour plus d'information, consulter les liens suivants :

- ▶ <https://pro.arcgis.com/fr/pro-app/latest/help/data/geocoding/about-locators.htm> ;
- ▶ <https://enterprise.arcgis.com/fr/server/latest/publish-services/windows/geocode-services.htm>  
(Voir chapitre « Créer un localisateur »).

! 1Spatial propose un mécanisme permettant d'utiliser un service de géocodage avec vos données mais ne fournit pas le service. Si nécessaire, rapprochez-vous de notre service commercial.

Le fichier archive contenant les données de géolocalisation utilisable par le mobile doit :

- ▶ se nommer « locator.zip » et être au format ZIP ;
- ▶ être déposé dans le répertoire « <Serveur 1Biz>\config\permanent\locator » avec création du répertoire si nécessaire ;
- ▶ Contenir deux fichiers obtenus à partir d'un service de géocodage dans ArcGIS Pro :
  - **locator.loz** : les données de géolocalisation ;
  - **locator.loc** : les options de recherche des données de géolocalisation.

Le fichier de configuration de la recherche par adresse a pour nom « locate\_by\_address.json » et se situe dans le répertoire « location », lui-même situé dans le répertoire « 1Biz ».

Ce fichier, au format JSON, a pour structure :

```
{
  "label": "Base d'adresses",
  "completionThreshold": 3,
  "hint": "Au moins 3 caractères. Exemple : boulevard du port",
  "inputType": "Address",
```

```
"maxFeatures": 30,  
"geographicRestriction": true  
}
```

Description des champs :

- ▶ **"label"** : Obligatoire. Le libellé affiché dans la liste des types de recherches ;
- ▶ **"completionThreshold"** : Facultatif. Le nombre de caractères à saisir avant qu'une recherche se déclenche (par défaut : 3) ;
- ▶ **"hint"** : Facultatif. Une courte description de la recherche. Par défaut : « Au moins *<completionThreshold>* caractères » ;
- ▶ **"inputType"** : Facultatif. Le type de clavier Android qui apparaîtra lors de la saisie des caractères de recherche. Les valeurs possibles sont :
  - **"Number"** : clavier numérique ;
  - **"Text"** : clavier alphanumérique standard (option par défaut) ;
  - **"Address"** : clavier orienté pour la saisie d'adresse.
- ▶ **"maxFeatures"** : Facultatif. Le nombre maximum de suggestion d'adresses dans la liste des adresses proposées (par défaut : 50) ;
- ▶ **"geographicRestriction"** : Facultatif. Permet de réduire la recherche aux adresses incluses dans la zone de travail de l'utilisateur (par défaut : true).

## 4.9 CONFIGURATION DU MODULE « GRAFFITI »

Le module « Graffiti » propose une liste de symboles à utiliser pour la saisie des ponctuels. Cet ensemble de symboles est regroupé dans un fichier « **icons.zip** », pour chaque application métier :

- ▶ dans le répertoire « *config/permanent/icons/onewater/water* » pour l'application Eau Potable ;
- ▶ dans le répertoire « *config/permanent/icons/onewater/sanitation* » pour l'application Assainissement.

Un fichier « icons.zip » est fourni pour chaque application avec l'installation standard mais il peut être modifié pour ajouter, mettre à jour ou supprimer des symboles.

Le fichier « icons.zip » de symboles d'une application métier, déposé sur le serveur, est téléchargé sur la tablette lors d'une synchronisation, si sa date de modification est plus récente que celle du fichier présent sur la tablette.

## 4.10 CONFIGURATION DE LA FONCTIONNALITÉ « GESTION DES CONSÉQUENCES »

Le fichier « **1Biz\consequences\display\_params.json** », optionnel, permet de configurer la couleur de mise en évidence des valeurs d'attributs modifiées, d'un élément du patrimoine, quand on consulte ou édite une modification en mobilité.

```
{  
  "highlightHintColor": "#FF0082AB"  
}
```

Figure 55 : Configuration de la gestion des conséquences en mobilité

- **hintHighlightColor** : Facultatif. Valeur au format hexadécimal de la couleur de mise en évidence des valeurs d'attributs modifiées (par défaut : **#FF0082AB**).

Dans l'exemple ci-dessous, le « Diamètre nominal » a été modifiée. Le champ et le groupe le contenant sont mis en évidence avec la couleur définie.



← E-Tronçon →

Description >

Caractéristiques v

Date de pose  
20 févr. 1956

Précision date de pose  
0

Matériau  
Fonte grise

Diamètre nominal (mm)  
140

Diamètre extérieur (mm)

**Figure 56 : Mise en évidence d'une conséquence**

## 5 UTILITAIRES

---

### 5.1 EXPORT EPANET

EPANET est un logiciel de modélisation d'écoulements hydrauliques des réseaux d'eau potable pour réaliser l'analyse de systèmes de distribution.

1Water propose un outil d'export du réseau d'eau potable, modélisé avec Utility Network, vers le format EPANET (inp) depuis la boîte à outils de 1Water Desktop et depuis le module « Export » de 1Water Web.

L'export EPANET de 1Water s'appuie sur un outil de Geoprocessing développé avec « ArcGIS API for Python ».

Ce chapitre décrit la syntaxe du fichier JSON de correspondance entre le modèle de données 1Water et le format attendu par EPANET. La partie utilisation n'est pas décrite dans ce document mais dans le guide utilisateur 1Water Desktop et le guide utilisateur 1Water Web.

#### 5.1.1 Fichier de paramétrage

Pour prendre en compte les différentes modélisations Utility Network d'un réseau d'eau potable 1Water, un fichier de paramétrage au format JSON assure la correspondance vers le format EPANET.

Le fichier de paramétrage est composé de plusieurs parties :

- ▶ **headers** : description des sections EPANET à exporter ;
- ▶ **domains** : description du contenu de chaque section à partir du modèle de données 1Water pour chaque domaine de réseau d'un Utility Network.
- ▶ **sections** : description des différentes sections déclarées dans la partie « headers » ;
- ▶ **dictionaries** : liste des dictionnaires utilisés dans les correspondances de valeurs entre 1Water et EPANET ;
- ▶ **modelatt** : description des modèles d'attributs pouvant être utilisés dans la partie « domains » ;
- ▶ **completeatt** : description de la succession d'attributs liée à un choix de valeur attributaire.

##### 5.1.1.1 Headers

Cette partie indique les sections à exporter avec les entêtes de colonne associées à chaque section.

L'ordre des sections dans cette partie indique l'ordre d'écriture dans le fichier résultat.

- ! L'ordre des sections importe peu. Néanmoins, un nœud ou un arc auquel se réfère une commande dans une section, doit être déjà défini dans les sections [JUNCTIONS], [RESERVOIRS], [TANKS], [PIPES], [PUMPS], ou [VALVES]. C'est pourquoi il est recommandé de mettre ces sections au début, juste au-dessous de la section [TITLE].

```
"headers": {
  "JUNCTIONS": [";ID Elevation"],
  "TANKS": [";ID Elevation InitLevel MinLevel MaxLevel Diameter MinVol"],
  "RESERVOIRS": [";ID Head"],
  "PIPES": [";ID Node1 Node2 Length Diameter Roughness MinorLoss Status"],
  "VALVES": [";ID Node1 Node2 Diameter Type Setting MinorLoss"],
  "PUMPS": [";ID Node1 Node2 Parameters"],
  "STATUS": [";ID Status/Setting"],
  "COORDINATES": [";Node X-Coord Y-Coord"],
  "VERTICES": [";Link X-Coord Y-Coord"]
},
```

Figure 57 : Partie « headers »

### 5.1.1.2 Options

Cette partie permet de définir la partie [OPTIONS] à déclarer dans le fichier d'export EPANet.

```
"options": [
  {
    "name": "UNITS",
    "value": "CMH"
  },
  {
    "name": "REQUIRED PRESSURE",
    "value": "40"
  },
  {
    "name": "HEADLOSS",
    "value": "H-W"
  }
],
```

Figure 58 : Partie « options »

### 5.1.1.3 Domains

Cette partie contient un tableau représentant les domaines, au sens Utility Network, pris en compte au niveau de 1Water pour réaliser l'export de données vers EPANET.

! Cette partie doit contenir au moins un domaine.

Un domaine est représenté par un nom (« name ») et par un tableau de sections (« [sections](#) »).

! Les sections indiquées dans ce tableau doivent être présentes dans la partie « headers ».

Chaque section contient la description des éléments de 1Water à exporter et à retranscrire vers le format EPANET.

```
"domains": [
  {
    "name": "RES_EAU",
    "sections": {
      "SYSTEM": [
        "JUNCTIONS": [
        "PIPES": [
        "VALVES": [
        "PUMPS": [
        "TANKS": [
        "RESERVOIRS": [
        "STATUS": [
      ]
    }
  }
],
```

Figure 59 : Partie "domains"

### 5.1.1.4 Sections

La partie « sections » décrit la conversion du modèle 1Water vers le modèle EPANET pour chaque section.



Une section est représentée par une liste d'entités du modèle 1Water au sens Utility Network.

Une entité est définie par :

- ▶ sa source : Line, Junction ou Device ;
- ▶ son groupe : code de l'assetgroup ;
- ▶ son type : code de l'assettype (-1 pour ne pas le prendre en compte dans le filtre).

```
"source": "Line",  
"group": 100,  
"type": 2,
```

Figure 60 : Entité d'une section

La section « SYSTEM » permet de gérer les jonctions « systèmes » du réseau de distribution.

```
"source": "System",  
"section": "JUNCTIONS",  
"group": -1,  
"type": -1,  
"filters": [],
```

Figure 61 : Section "SYSTEM"

Une entité est éventuellement associée à un ou plusieurs filtres attributaires (« filters »). Un filtre attributaire est défini par un attribut du modèle 1Water (« attribute »), un opérateur de comparaison (« ope ») et une valeur de comparaison (« value »).

```
"filters": [  
  {  
    "attribute": "fonction",  
    "ope": "==",  
    "value": "distribution"  
  }  
],
```

Figure 62 : Filtre sur une entité

Les opérateurs de comparaison possibles sont :

- ▶ « == » : égal à ;
- ▶ « != » : différent de ;
- ▶ « < » : strictement inférieur à ;
- ▶ « > » : strictement supérieur à ;
- ▶ « <= » : inférieur ou égal à ;
- ▶ « >= » : supérieur ou égal à.

Une entité est ensuite associée à une liste de définitions de valeur « values » pour renseigner chaque colonne déclarée dans la partie « headers » de la section.

La liste de définitions de valeur est soit déclarée directement dans le groupe « values », soit le groupe « values » est associé à un de [modèle d'attributs « modelatt »](#).

Une valeur est représentée par :

- ▶ **attribute** : nom de l'attribut 1Water ;
- ▶ **format** : format texte d'écriture de la valeur de l'attribut au sens Python ;
- ▶ **dictionary** : nom du dictionnaire, défini dans la partie « [dictionaries](#) », pour convertir une valeur (optionnel) ;
- ▶ **complete** : nom du groupe de complément d'attribut dans la partie « [completeatt](#) » (optionnel) ;
- ▶ **formule** : chaîne décrivant une formule utilisable avec la méthode « eval » du langage Python (optionnel) ;

- **ope** : opération de comparaison associée à « value » (optionnel) ;
- **value** : valeur à comparer avec celle de « attribute » en utilisant l'opération « ope » (optionnel) ;
- **next** : permet de définir un autre attribut 1Water à lire si « attribute » n'est pas renseigné (optionnel) ou si la condition « attribute » avec « ope » et « value » est vrai.  
Dans « next », il est possible de lister plusieurs attributs à lire successivement tant qu'aucune valeur n'est récupérée.

La définition d'un bloc « next », pour chaque attribut, contient :

- **attribute** : nom de l'attribut 1Water ;
  - **ope** : opération de comparaison associée à « value » (optionnel) ;
  - **value** : valeur à comparer avec celle de « attribute » en utilisant l'opération « ope » (optionnel) ;
- **default** : valeur à utiliser si « attribute » ou le résultat de « next » ne donne pas de résultat.

```
{
  "attribute": "diametre",
  "format": "%d",
  "dictionary": "Dico_Diametre_Tronçon",
  "next": [
    {
      "attribute": "diamext"
    }
  ],
  "default": 100
},
```

Figure 63 : Valeur avec dictionnaire et « next »

```
{
  "formule": "",
  "attribute": "diametre",
  "format": "%d",
  "ope": "<=",
  "value": 0,
  "dictionary": "",
  "next": [
    {
      "attribute": "hauteur",
      "ope": "<=",
      "value": 0
    },
    {
      "attribute": "largeur",
      "ope": "<=",
      "value": 0
    }
  ],
  "default": 100
},
```

Figure 64 : Valeur opération de comparaison et plusieurs « next »

### 5.1.1.5 Dictionaries

Cette partie permet de définir les dictionnaires utilisés pour réaliser des correspondances de valeur entre 1Water et EPANET.

Chaque dictionnaire est défini par un nom (« name ») et un tableau de correspondances (« matching »).

Chaque correspondance donne la valeur EPANET (« value ») associée à une valeur 1Water (« key ») par rapport à un opérateur de comparaison (« ope »). Si l'opérateur de comparaison n'est pas déclaré, c'est une comparaison par égalité qui est réalisée.

Les opérateurs possibles sont :

- ▶ « == » : égal à ;
- ▶ « != » : différent de ;
- ▶ « < » : strictement inférieur à ;
- ▶ « > » : strictement supérieur à ;
- ▶ « <= » : inférieur ou égal à ;
- ▶ « >= » : supérieur ou égal à .

Pour la gestion d'une valeur quelconque de « key », l'opérateur de comparaison prend la valeur « \* ». Dans ce cas, quelle que soit la valeur de provenant de 1Water (« key »), la valeur EPANET prend « value ». Quand l'opérateur de comparaison vaut « \* », la balise « key » n'est pas prise en compte.

```
"dictionaries": [  
  {  
    "name": "Dico_Position_Vanne",  
    "matching": [  
      {  
        "key": -1,  
        "value": "OPEN"  
      },  
      {  
        "key": 0,  
        "value": "OPEN"  
      },  
      {  
        "key": 701,  
        "value": "OPEN"  
      },  
      {  
        "key": 702,  
        "value": "CLOSED"  
      },  
      {  
        "key": 703,  
        "value": "OPEN"  
      },  
      {  
        "key": 704,  
        "value": "CLOSED"  
      }  
    ]  
  },  
]
```

Figure 65 : Exemple de dictionnaire avec un opérateur de comparaison d'égalité implicite

```
{  
  "name": "Dico_Non_negatif",  
  "matching": [  
    {  
      "key": 0,  
      "ope": "<",  
      "value": 0  
    }  
  ]  
},
```

Figure 66 : Exemple de dictionnaire avec un opérateur de comparaison strictement inférieur

La gestion des intervalles est réalisée en utilisant plusieurs correspondances de valeurs successives basées sur des opérateurs de comparaison. La première correspondance trouvée est prise en compte.

### 5.1.1.6 Attributs

Cette partie permet de déclarer des modèles d'informations attributaires partagés par plusieurs sections des « domains ».

#### 5.1.1.6.1 Groupe « modelatt »

La déclaration d'un modèle d'attributs « modelatt » permet d'éviter de déclarer plusieurs fois la description de valeurs d'attribut si elle est partagée par plusieurs sections.

Un modèle d'attributs est défini par un nom (« name ») et associé à une liste de définitions de valeur (« values »). Le nom d'un modèle d'attributs est associé à une balise « values » d'une section.

La description de la liste de définitions de valeur est la même que celle [décrite dans la « sections »](#).

```
{
  "name": "att_OUTLETS_FUNCTIONAL",
  "values": [
    {
      "formule": "",
      "attribute": "cotetn",
      "format": "",
      "dictionary": "",
      "next": [],
      "default": 0
    },
    {
      "formule": "",
      "attribute": "radier",
      "format": "",
      "dictionary": "",
      "next": [],
      "default": 0
    },
    {
      "formule": "cotetn - radier",
      "attribute": "",
      "format": "%0.2f",
      "dictionary": "",
      "next": [],
      "complete": "att_OUTLETS_offset",
      "default": 1.0
    },
    {
      "formule": "",
      "attribute": "flapgate",
      "format": "%s",
      "dictionary": "",
      "next": [],
      "default": "NO"
    }
  ]
},
```

Figure 67 : Exemple d'un modèle d'attributs

#### 5.1.1.6.2 Groupe « completeatt »

Un modèle conditionnel d'attributs est défini par un nom (« name ») et une liste de conditions. Le nom d'un modèle conditionnel d'attributs est associé à une balise « complete » d'une définition de valeur.

La liste de conditions est définie par une valeur conditionnelle (« value ») associé à une liste de définitions de valeur (« values »).

La valeur conditionnelle est comparée à la valeur retournée par « attribute » ou « formule » de la définition de valeur parent.

La description de la liste de définitions de valeur est la même que celle [décrite dans la « sections »](#).

```
"completeatt": [
  {
    "name": "att_VALVES_type",
    "matching": [
      {
        "value": "PSV",
        "values": [
          {
            "formule": "",
            "attribute": "pressamont_valves",
            "format": "%0.2f",
            "dictionary": "",
            "next": [],
            "default": 100.0
          }
        ]
      },
      {
        "value": "FCV",
        "values": [
          {
            "formule": "",
            "attribute": "debit_valves",
            "format": "%0.2f",
            "dictionary": "",
            "next": [],
            "default": 100.0
          }
        ]
      },
      {
        "value": "TCV",
        "values": [
          {
            "formule": "",
            "attribute": "coefcharge_valves",
            "format": "%0.2f",
            "dictionary": "",
            "next": [],
            "default": 1.0
          }
        ]
      }
    ]
  }
],
```

Figure 68 : Exemple d'un modèle conditionnel d'attributs

#### 5.1.1.6.3 Paramètres

Dans la description d'une définition de valeur, la balise « format » permet d'indiquer le format d'écriture d'une valeur dans le fichier résultat SWMM. Dans certains cas, il peut être intéressant de récupérer une valeur sans pour autant l'écrire dans le résultat. Si la balise « format » n'est pas renseignée, la valeur est conservée en mémoire sous le nom de l'étiquette « attribute » pour pouvoir l'utiliser par la suite.

L'exemple ci-dessous indique comment écrire dans le fichier résultat la profondeur calculée par la différence entre la valeur de l'attribut « Cote TN » et celle de l'attribut « Radier ».

```
{
  "name": "att_OUTLETS_FUNCTIONAL",
  "values": [
    {
      "formule": "",
      "attribute": "cotetn",
      "format": "",
      "dictionary": "",
      "next": [],
      "default": 0
    },
    {
      "formule": "",
      "attribute": "radier",
      "format": "",
      "dictionary": "",
      "next": [],
      "default": 0
    },
    {
      "formule": "cotetn - radier",
      "attribute": "",
      "format": "%0.2f",
      "dictionary": "",
      "next": [],
      "complete": "att_OUTLETS_offset",
      "default": 1.0
    },
    {
      "formule": "",
      "attribute": "flapgate",
      "format": "%s",
      "dictionary": "",
      "next": [],
      "default": "NO"
    }
  ]
},
```

**Figure 69 : Partie « Paramètres »**

## 5.1.2 Exemple

```
"PIPES": [
  {
    "source": "Line",
    "group": 100,
    "type": 2,
    "filters": [],
    "values": [
      {
        "attribute": "longueur",
        "format": "%0.2f",
        "dictionary": "",
        "next": [],
        "default": 0
      },
      {
        "attribute": "diametre",
        "format": "%d",
        "dictionary": "Dico_Diametre_Tronçon",
        "next": [
          {
            "attribute": "diamext"
          }
        ],
        "default": 100
      },
      {
        "attribute": "rugosite",
        "format": "%d",
        "dictionary": "Dico_Rugosite",
        "next": [],
        "default": 100
      },
      {
        "attribute": "",
        "format": "%d",
        "dictionary": "",
        "next": [],
        "default": 0
      },
      {
        "attribute": "",
        "format": "%s",
        "dictionary": "Dico_Etat_Troncon",
        "next": [],
        "default": 1
      }
    ]
  }
],
```

Figure 70 : Exemple de la section « PIPES »

## 5.2 EXPORT SWMM

SWMM est un logiciel de modélisation de gestion des eaux pluviales et unitaires, ainsi que des pluies sur les bassins versants et des injections d'eau représentant les eaux usées.

1Water propose un outil d'export du réseau des eaux pluviales, modélisé avec Utility Network, vers le format SWMM (inp) depuis la boîte à outils de 1Water Desktop et depuis le module « Export » de 1Water Web.

L'export SWMM de 1Water s'appuie sur un outil de Geoprocessing développé avec « ArcGIS API for Python ».

Ce chapitre décrit la syntaxe du fichier JSON de correspondance entre le modèle de données 1Water et le format attendu par SWMM. La partie utilisation n'est pas décrite dans ce document mais dans le guide utilisateur 1Water Desktop et le guide utilisateur 1Water Web.

## 5.2.1 Fichier de paramétrage

Pour prendre en compte les différentes modélisations Utility Network d'un réseau d'eau potable 1Water, un fichier de paramétrage au format JSON assure la correspondance vers le format SWMM.

Le fichier de paramétrage est composé de plusieurs parties :

- ▶ **headers** : description des sections SWMM à exporter ;
- ▶ **options** : description des options de SWMM ;
- ▶ **domains** : description du contenu de chaque section à partir du modèle de données 1Water pour chaque domaine de réseau d'un Utility Network.
- ▶ **sections** : description des différentes sections déclarées dans la partie « headers » ;
- ▶ **dictionaries** : liste des dictionnaires utilisés dans les correspondances de valeurs entre 1Water et SWMM ;
- ▶ **modelatt** : description des modèles d'attributs pouvant être utilisés dans la partie « domains » ;
- ▶ **completeatt** : description de la succession d'attributs liée à un choix de valeur attributaire.

### 5.2.1.1 Headers

Cette partie indique les sections à exporter avec les entêtes de colonne associées à chaque section.

L'ordre des sections dans cette partie indique l'ordre d'écriture dans le fichier résultat.

- ! L'ordre des sections importe peu. Néanmoins, un nœud ou un arc auquel se réfère une commande dans une section, doit être déjà défini dans les sections [JUNCTIONS], [OUTFALLS], [DIVIDERS], [STORAGE], [ORIFICES], [WEIRS], [PUMPS], ou [OUTLETS]. C'est pourquoi il est recommandé de mettre ces sections au début, juste au-dessous de la section [TITLE].

```
"headers": {
  "JUNCTIONS": [":ID Elev Ymax Y0 Ysur Apond"],
  "OUTFALLS": [
    ":ID Elev FREE Gated RouteTo",
    ":ID Elev NORMAL Gated RouteTo",
    ":ID Elev FIXED Stage Gated RouteTo",
    ":ID Elev TIDAL Tcurve Gated RouteTo",
    ":ID Elev TIMESERIES Tseries Gated RouteTo"
  ],
  "DIVIDERS": [
    ":ID Elev DivertedLink OVERFLOW Ymax Y0 Ysur Apond",
    ":ID Elev DivertedLink CUTOFF Qmin Ymax Y0 Ysur Apond",
    ":ID Elev DivertedLink TABULAR Dcurve Ymax Y0 Ysur Apond",
    ":ID Elev DivertedLink WEIR Qmin Ht Cd Ymax Y0 Ysur Apond"
  ],
  "STORAGE": [
    ":ID Elev Ymax Y0 TABULAR Acurve",
    ":ID Elev Ymax Y0 FUNCTIONAL A1 A2 A0",
    ":ID Elev Ymax Y0 Shape L W Z"
  ],
  "CONDUITS": [":ID InletNode OutletNode Length Roughness Z1 Z2 Q0 Qmax"],
  "XSECTIONS": [
    ":Link Shape Geom1 Geom2 Geom3 Geom4 Barrels Culvert",
    ":Link CUSTOM Geom1 Curve Barrels",
    ":Link IRREGULAR Tsect",
    ":Link STREET Street"
  ],
  "PUMPS": [":ID InletNode OutletNode Pcurve Status Startup Shutoff"],
  "ORIFICES": [":ID InletNode OutletNode Type Offset Cd Flap Orate"],
  "WEIRS": [":ID InletNode OutletNode Type CrstHt Cd Gated EC Cd2 Sur Width Surf"],
  "OUTLETS": [
    ":ID InletNode OutletNode Offset TABULAR/DEPTH Qcurve Gated",
    ":ID InletNode OutletNode Offset TABULAR/HEAD Qcurve Gated",
    ":ID InletNode OutletNode Offset FUNCTIONAL/DEPTH C1 C2 Gated",
    ":ID InletNode OutletNode Offset FUNCTIONAL/HEAD C1 C2 Gated"
  ],
  "COORDINATES": [":Node X-Coord Y-Coord"],
  "VERTICES": [":Link X-Coord Y-Coord"]
},
```

Figure 71 : Partie « headers »

### 5.2.1.2 Options

Cette partie permet de définir la partie [OPTIONS] à déclarer dans le fichier d'export SWMM.



```
"options": [  
  {  
    "name": "THREADS",  
    "value": "1"  
  },  
  {  
    "name": "FLOW_UNITS",  
    "value": "CFS"  
  },  
  {  
    "name": "LINK_OFFSETS",  
    "value": "DEPTH"  
  },  
  {  
    "name": "START_TIME",  
    "value": "00:00:00"  
  },  
  {  
    "name": "END_TIME",  
    "value": "23:59:59"  
  }  
],
```

Figure 72 : Partie "options"

### 5.2.1.3 Domains

Cette partie contient un tableau représentant les domaines, au sens Utility Network, pris en compte au niveau de 1Water pour réaliser l'export de données vers SWMM.

! Cette partie doit contenir au moins un domaine.

Un domaine est représenté par un nom (« name ») et par un tableau de sections (« [sections](#) »).

! Les sections indiquées dans ce tableau doivent être présentes dans la partie « headers ».

Chaque section contient la description des éléments de 1Water à exporter et à retranscrire vers le format SWMM.

```
"domains": [  
  {  
    "name": "RES_ASS",  
    "sections": {  
      "SYSTEM": [  
        "JUNCTIONS": [  
          "OUTFALLS": [  
            "DIVIDERS": [],  
            "XSECTIONS": [  
              "PUMPS": [  
                "ORIFICES": [  
                  "WEIRS": [  
                    "OUTLETS": [  
                      "STORAGE": [  
                        "CONDUITS": [  
                          ]  
                        ]  
                      ]  
                    ]  
                  ]  
                ]  
              ]  
            ]  
          ]  
        ]  
      ]  
    }  
  }  
]
```

Figure 73 : Partie "domains"

### 5.2.1.4 Sections

La partie « sections » décrit la conversion du modèle 1Water vers le modèle SWMM pour chaque section.

Une section est représentée par une liste d'entités du modèle 1Water au sens Utility Network.

Une entité est définie par :

- ▶ sa source : Line, Junction ou Device ;
- ▶ son groupe : code de l'assetgroup ;
- ▶ son type : code de l'assettype (-1 pour ne pas le prendre en compte dans le filtre).

```
"source": "Line",  
"group": 200,  
"type": -1,
```

Figure 74 : Entité d'une section

La section « SYSTEM » permet de gérer les jonctions « systèmes » du réseau de distribution.

```
"source": "System",  
"section": "JUNCTIONS",  
"group": -1,  
"type": -1,  
"filters": [],
```

Figure 75 : Section "SYSTEM"

Une entité est éventuellement associée à un ou plusieurs filtres attributaires (« filters »). Un filtre attributaire est défini par un attribut du modèle 1Water (« attribute »), un opérateur de comparaison (« ope ») et une valeur de comparaison (« value »).

```
"filters": [  
  {  
    "attribute": "fonction",  
    "ope": "==",  
    "value": "distribution"  
  }  
],
```

Figure 76 : Filtre sur une entité

Les opérateurs de comparaison possibles sont :

- ▶ « == » : égal à ;
- ▶ « != » : différent de ;
- ▶ « < » : strictement inférieur à ;
- ▶ « > » : strictement supérieur à ;
- ▶ « <= » : inférieur ou égal à ;
- ▶ « >= » : supérieur ou égal à.

Une entité est ensuite associée à une liste de définitions de valeur « values » pour renseigner chaque colonne déclarée dans la partie « headers » de la section.

La liste de définitions de valeur est soit déclarée directement dans le groupe « values » soit le groupe « values » est associé à un de [modèle d'attributs « modelatt »](#).

Une valeur est représentée par :

- ▶ **attribute** : nom de l'attribut 1Water ;
- ▶ **format** : format texte d'écriture de la valeur de l'attribut au sens Python ;
- ▶ **dictionary** : nom du dictionnaire, défini dans la partie « [dictionaries](#) », pour convertir une valeur (optionnel) ;
- ▶ **complete** : nom du groupe de complément d'attribut dans la partie « [completeatt](#) » (optionnel) ;
- ▶ **formule** : chaîne décrivant une formule utilisable avec la méthode « eval » du langage Python (optionnel) ;
- ▶ **ope** : opération de comparaison associée à « value » (optionnel) ;
- ▶ **value** : valeur à comparer avec celle de « attribute » en utilisant l'opération « ope » (optionnel) ;

- **next** : permet de définir un autre attribut 1Water à lire si « attribute » n'est pas renseigné (optionnel) ou si la condition « attribute » avec « ope » et « value » est vrai.  
Dans « next », il est possible de lister plusieurs attributs à lire successivement tant qu'aucune valeur n'est récupérée.  
La définition d'un bloc « next », pour chaque attribut, contient :
  - **attribute** : nom de l'attribut 1Water ;
  - **ope** : opération de comparaison associée à « value » (optionnel) ;
  - **value** : valeur à comparer avec celle de « attribute » en utilisant l'opération « ope » (optionnel) ;
- **default** : valeur à utiliser si « attribute » ou le résultat de « next » ne donne pas de résultat.

```
{
  "attribute": "diametre",
  "format": "%d",
  "dictionary": "Dico_Diametre_Tronçon",
  "next": [
    {
      "attribute": "diamext"
    }
  ],
  "default": 100
},
```

Figure 77 : Valeur avec dictionnaire et « next »

```
{
  "formule": "",
  "attribute": "diametre",
  "format": "%d",
  "ope": "<=",
  "value": 0,
  "dictionary": "",
  "next": [
    {
      "attribute": "hauteur",
      "ope": "<=",
      "value": 0
    },
    {
      "attribute": "largeur",
      "ope": "<=",
      "value": 0
    }
  ],
  "default": 100
},
```

Figure 78 : Valeur opération de comparaison et plusieurs « next »

### 5.2.1.5 Dictionaries

Cette partie permet de définir les dictionnaires utilisés pour réaliser des correspondances de valeur entre 1Water et SWMM.

Chaque dictionnaire est défini par un nom (« name ») et un tableau de correspondances (« matching »).

Chaque correspondance donne la valeur SWMM (« value ») associée à une valeur 1Water (« key ») par rapport à un opérateur de comparaison (« ope »). Si l'opérateur de comparaison n'est pas déclaré, c'est une comparaison par égalité qui est réalisée.

Les opérateurs possibles sont :

- « == » : égal à ;
- « != » : différent de ;
- « < » : strictement inférieur à ;
- « > » : strictement supérieur à ;
- « <= » : inférieur ou égal à ;

► « >= » :supérieur ou égal à.

Pour la gestion d'une valeur quelconque de « key », l'opérateur de comparaison prend la valeur « \* ». Dans ce cas, quelle que soit la valeur de provenant de 1Water (« key »), la valeur SWMM prend « value ». Quand l'opérateur de comparaison vaut « \* », la balise « key » n'est pas prise en compte.

```
"dictionaries": [
  {
    "name": "Dico_forme_XSECTION",
    "matching": [
      {
        "key": 0,
        "value": "RECT_CLOSED"
      },
      {
        "key": 99,
        "value": "RECT_CLOSED"
      },
      {
        "key": 1,
        "value": "CIRCULAR"
      },
      {
        "key": 2,
        "value": "RECT_CLOSED"
      },
      {
        "key": 3,
        "value": "EGG"
      },
      {
        "key": 4,
        "value": "PARABOLIC"
      },
      {
        "key": 5,
        "value": "ARCH"
      },
      {
        "key": 6,
        "value": "HORIZ_ELLIPSE"
      },
      {
        "key": -1,
        "ope": "OK",
        "value": "RECT_CLOSED"
      }
    ]
  }
],
```

Figure 79 : Exemple de dictionnaire avec un opérateur de comparaison d'égalité implicite

```
{
  "name": "Dico_Non_negatif",
  "matching": [
    {
      "key": 0,
      "ope": "<",
      "value": 0
    }
  ]
},
```

Figure 80 : Exemple de dictionnaire avec un opérateur de comparaison strictement inférieur

La gestion des intervalles est réalisée en utilisant plusieurs correspondances de valeurs successives basées sur des opérateurs de comparaison. La première correspondance trouvée est prise en compte.

## 5.2.1.6 Attributs

Cette partie permet de déclarer des modèles d'informations attributaires partagés par plusieurs sections des « domains ».

### 5.2.1.6.1 Groupe « modelatt »

La déclaration d'un modèle d'attributs « modelatt » permet d'éviter de déclarer plusieurs fois la description de valeurs d'attribut si elle est partagée par plusieurs sections.

Un modèle d'attributs est défini par un nom (« name ») et associé à une liste de définitions de valeur (« values »). Le nom d'un modèle d'attributs est associé à une balise « values » d'une section.

La description de la liste de définitions de valeur est la même que celle [décrite dans la « sections »](#).

```
{
  "name": "att_OUTLETS_FUNCTIONAL",
  "values": [
    {
      "formule": "",
      "attribute": "cotetn",
      "format": "",
      "dictionary": "",
      "next": [],
      "default": 0
    },
    {
      "formule": "",
      "attribute": "radier",
      "format": "",
      "dictionary": "",
      "next": [],
      "default": 0
    },
    {
      "formule": "cotetn - radier",
      "attribute": "",
      "format": "%0.2f",
      "dictionary": "",
      "next": [],
      "complete": "att_OUTLETS_offset",
      "default": 1.0
    },
    {
      "formule": "",
      "attribute": "flapgate",
      "format": "%s",
      "dictionary": "",
      "next": [],
      "default": "NO"
    }
  ]
},
```

**Figure 81 : Exemple d'un modèle d'attributs**

#### 5.2.1.6.2 Groupe « completeatt »

Un modèle conditionnel d'attributs est défini par un nom (« name ») et une liste de conditions. Le nom d'un modèle conditionnel d'attributs est associé à une balise « complete » d'une définition de valeur.

La liste de conditions est définie par une valeur conditionnelle (« value ») associé à une liste de définitions de valeur (« values »).

La valeur conditionnelle est comparée à la valeur retournée par « attribute » ou « formule » de la définition de valeur parent.

La description de la liste de définitions de valeur est la même que celle [décrite dans la « sections »](#).

```
{
  "name": "att_OUTFALLS_type",
  "matching": [
    {
      "value": "FIXED",
      "values": [
        {
          "formule": "",
          "attribute": "stage_outfall",
          "format": "%0.2f",
          "dictionary": "",
          "next": [],
          "default": 1.0
        }
      ]
    },
    {
      "value": "TIDAL",
      "values": [
        {
          "formule": "",
          "attribute": "rcurv_outfall",
          "format": "%s",
          "dictionary": "",
          "next": [],
          "default": ""
        }
      ]
    },
    {
      "value": "TIMESERIES",
      "values": [
        {
          "formule": "",
          "attribute": "timeseries_outfall",
          "format": "%s",
          "dictionary": "",
          "next": [],
          "default": ""
        }
      ]
    }
  ]
},
```

Figure 82 : Exemple d'un modèle conditionnel d'attributs

#### 5.2.1.6.3 Paramètres

Dans la description d'une définition de valeur, la balise « format » permet d'indiquer le format d'écriture d'une valeur dans le fichier résultat SWMM. Dans certains, il peut être intéressant de récupérer une valeur sans pour autant l'écrire dans le résultat. Si la balise « format » n'est pas renseignée, la valeur est conservée en mémoire sous le nom de l'étiquette « attribute » pour pouvoir l'utiliser par la suite.

L'exemple ci-dessous indique comment écrire dans le fichier résultat la profondeur calculée par la différence entre la valeur de l'attribut « Cote TN » et celle de l'attribut « Radier ».

```
{
  "name": "att_OUTLETS_FUNCTIONAL",
  "values": [
    {
      "formule": "",
      "attribute": "cotetn",
      "format": "",
      "dictionary": "",
      "next": [],
      "default": 0
    },
    {
      "formule": "",
      "attribute": "radier",
      "format": "",
      "dictionary": "",
      "next": [],
      "default": 0
    },
    {
      "formule": "cotetn - radier",
      "attribute": "",
      "format": "%0.2f",
      "dictionary": "",
      "next": [],
      "complete": "att_OUTLETS_offset",
      "default": 1.0
    },
    {
      "formule": "",
      "attribute": "flapgate",
      "format": "%s",
      "dictionary": "",
      "next": [],
      "default": "NO"
    }
  ]
},
```

Figure 83 : Partie « Paramètres »

## 5.2.2 Exemple

```
"domains": [
  {
    "name": "RES_EP",
    "sections": {
      "OUTFALLS": [
        {
          "source": "Device",
          "group": 262,
          "type": -1,
          "filters": [],
          "values": "att_OUTFALLS"
        }
      ]
    }
  },
]
```

Figure 84 : Exemple de la section « OUTFALLS »

```
"modelatt": [
  {
    "name": "att_OUTFALLS",
    "values": [
      {
        "formule": "",
        "attribute": "radier",
        "format": "%0.2f",
        "dictionary": "",
        "next": [],
        "complete": "",
        "default": 0.0
      },
      {
        "formule": "",
        "attribute": "type_outfalls",
        "format": "%s",
        "dictionary": "",
        "next": [],
        "complete": "att_OUTFALLS_type",
        "default": "FREE"
      },
      {
        "formule": "",
        "attribute": "flapgate",
        "format": "%s",
        "dictionary": "",
        "next": [],
        "default": "NO"
      },
      {
        "formule": "",
        "attribute": "routeto",
        "format": "%s",
        "dictionary": "",
        "next": [],
        "default": ""
      }
    ]
  }
],
```

Figure 85 : Exemple de modèle d'attributs « OUTFALLS »

## 5.3 EXPORT STAR-DT

Ce chapitre intervient après une révision de la réglementation anti-endommagement, mise en place dès 2012, qui vise à réduire les dommages causés aux réseaux lors de travaux effectués dans leur voisinage et à prévenir leurs conséquences néfastes pour la sécurité des personnes et des biens mais aussi la protection de l'environnement.

Sur le [site du CNIG](#) :

- ▶ Arrêté du 15 février 2012 sur Légifrance ;
- ▶ Arrêté du 26 octobre 2018 sur Légifrance.

Arrêté du 26 octobre 2018 portant modification de plusieurs arrêtés relatifs à l'exécution de travaux à proximité des réseaux et approbation d'une version modifiée des prescriptions techniques prévues à l'article R. 554-29 du code de l'environnement.

Le standard PCRS concerne désormais tous les ouvrages, en toutes localisations, avant le 1er janvier 2026.

« Le 7° du I de l'article 7 est applicable à tous les ouvrages, sensibles et non sensibles, dès l'existence effective dans la zone géographique concernée du lever régulier à grande échelle mentionné dans cet article, et au plus tard le 1er janvier 2026. »

Rappel du 7° du I de l'article 7 : « Le fond de plan employé pour la transmission des données de localisation des réseaux aux déclarants est le meilleur lever régulier à grande échelle disponible, établi



et mis à jour par l'autorité publique locale compétente en conformité avec les articles L. 127-1 et suivants du code de l'environnement et selon le format d'échange PCRS (plan corps de rue simplifié) établi et mis à jour par le Conseil national de l'information géographique.»

De plus, les 6° du I de l'article 7 et les articles 7-1 et 7-2 de l'arrêté du 15 février 2012 sont applicables :

- ▶ Le 1er janvier 2020 aux ouvrages souterrains sensibles pour la sécurité implantée dans des unités urbaines INSEE ;
- ▶ Le 1er janvier 2026 à tous les ouvrages souterrains sensibles pour la sécurité implantée sur l'ensemble du territoire ainsi qu'aux ouvrages souterrains non-sensibles implantés dans des unités urbaines au sens de l'INSEE ;
- ▶ Le 1er janvier 2032 à tous les ouvrages souterrains implantés sur l'ensemble du territoire.

Voir aussi la page du CNIG [GT Réseaux / GP4 DTDICT](#) sur le Groupe de travail cartes des réseaux enterrés.

### 5.3.1 Définition

**Standard géographique de Réseaux pour les réponses aux Déclarations de Travaux** : Le StaR-DT est un géostandard de description simplifiée des réseaux relevant de la réglementation anti-endommagement, à savoir l'arrêté du 15 février 2012 modifié en application du décret DT-DICT.

Limité aux objets les plus utiles et en n'abordant aucune logique « métiers », le StaR-DT constitue le socle commun de base décrivant à très grande échelle les réseaux concernés par le Décret DT-DICT afin de permettre à leurs gestionnaires et aux collectivités concernées d'échanger les informations utiles à la géolocalisation des ouvrages en amont des travaux.

Ce géostandard n'a pas vocation à être utilisé pour spécifier des bases de données métiers pour les exploitants de réseaux

C'est un **format standard d'échanges en vue d'une réponse DT-DICT**.

### 5.3.2 Exporter les données du réseau au format StaR-DT

C'est le cas d'utilisation par excellence du StaR-DT au sens des données échangées : l'export doit si possible être indépendant de l'outillage actuel des acteurs concernés, et prévoir par ailleurs des solutions quant à la vérification de la conformité au modèle des données échangées.

### 5.3.3 Modèle conceptuel de données

Le modèle conceptuel général StaR-DT repose sur un ensemble d'éléments de réseaux génériques qui regroupent les éléments physiques du réseau :

- ▶ Les tronçons qui regroupent les éléments "linéaires" du réseau ;
- ▶ Les nœuds qui regroupent les éléments ponctuels constitutifs du réseau ;
- ▶ Les conteneurs de nœud qui regroupent les éléments supports au réseau ;
- ▶ Les enveloppes d'arc qui regroupe les tronçons constituant une construction dans laquelle les câbles et les canalisations sont protégés et guidés.

Des éléments supplémentaires, liés aux éléments du réseau viennent ajouter des informations additionnelles.

Ces éléments sont à disposition de l'exploitant de réseau pour lui permettre de décrire ses ouvrages de la manière la plus appropriée et automatisée possible pour répondre à ses obligations mentionnées dans les cas d'usage.

Il choisit parmi les différentes possibilités de modélisation offertes celles qui lui conviennent le mieux, mais il n'est en aucun cas dans l'obligation de renseigner l'ensemble des informations potentiellement modélisables (à l'exception des attributs réglementaires obligatoires).

### 5.3.4 Correspondance 1Water

1Water propose un outil d'export des réseaux « Eau » et « Assainissement », modélisé avec Utility Network, vers le format StaR-DT en fichiers ShapeFile. Cet export est réalisable depuis la boîte à outils de 1Water Desktop et depuis le module « Export » de 1Water Web.

L'export StaR-DT de 1Water s'appuie sur un outil de Geoprocessing développé avec « ArcGIS API for Python ».

Ce document décrit la syntaxe du fichier JSON de correspondance entre le modèle de données 1Water et le format attendu par StaR-DT.

! La partie utilisation n'est pas décrite dans ce document mais dans le guide utilisateur 1Water Desktop et le guide utilisateur 1Water Web.

### 5.3.5 Fichier de paramétrage

Pour prendre en compte les différentes modélisations Utility Network d'un réseau « Eau » ou « Assainissement » 1Water, un fichier de paramétrage au format JSON assure la correspondance vers le format StaR-DT.

Le fichier de paramétrage est composé de plusieurs parties :

- ▶ **headers** : description des attributs des sections de données StaR-DT à exporter ;
- ▶ **parameters** : déclaration de la section des éléments servant à faire le lien entre 1Water et StaR-DT ;
- ▶ **dictionaries** : liste des dictionnaires utilisés dans les correspondances de valeurs entre 1Water et StaR-DT ;
- ▶ **domains** : description du contenu de chaque attribut à partir du modèle de données 1Water pour chaque domaine de réseau d'un Utility Network.
- ▶ **sections** : description des différentes sections déclarées dans la partie « headers » ;
- ▶ **modelatt** : description des modèles d'attributs pouvant être utilisés dans la partie « domains » .

#### 5.3.5.1 Headers

Cette partie indique les sections à exporter avec le descriptif des attributs à exporter pour chaque section.

```
"headers": {
  "Accessoire": {
    "CanalisationAssainissementPluviale": {
    "CanalisationEau": {
    "RegardNonVisitable": {
    "RegardVisitable": {
  },
},
```

**Figure 86 : Partie « headers »**

! Les sections doivent correspondre au contenu de la Géodatabase fichiers d'export qui est utilisée par le traitement.

Pour chaque section, un descriptif des champs « StaR-DT » est défini avec un nom associé à son type, à sa taille maximale ou son format.

```
"headers": {
  "Accessoire": {
    "id": [
      "Text",
      10
    ],
    "code": [
      "Text",
      80
    ],
    "precisioXY": [
      "Text",
      1
    ],
    "precisionZ": [
      "Text",
      1
    ],
    "visibleS": [
      "Short",
      1
    ],
    "assainissement": [

```

**Figure 87 : Contenu de « headers »**

### 5.3.5.2 Parameters

Permet la description des éléments à partir desquels le lien entre la base « 1Water » et les données « StaR-DT » est réalisé pour chaque nom de domaine.

```
"parameters": {
  "domains": {
    "RES_EAU": {
      "link": [
        "CanalisationEau"
      ]
    },
    "RES_EU": {
      "link": [
        "CanalisationAssainissementPluviale"
      ]
    },
    "RES_EP": {
      "link": [
        "CanalisationAssainissementPluviale"
      ]
    }
  }
},
},
```

**Figure 88 : Partie « Paramètres »**

### 5.3.5.3 Dictionaries

Cette partie permet de définir les dictionnaires utilisés pour réaliser les correspondances entre « 1Water » et « StaR-DT ».

Chaque dictionnaire est défini par un nom (« name ») et un tableau de correspondances (« matching »).

Chaque correspondance donne la valeur « StaR-DT » (« value ») associée à une valeur « 1Water » (« key ») par rapport à un opérateur de comparaison (« ope »). Si l'opérateur de comparaison n'est pas déclaré, c'est une comparaison par égalité qui est réalisée.

Les opérateurs possibles sont :

- ▶ « == » : égal à ;
- ▶ « != » : différent de ;
- ▶ « < » : strictement inférieur à ;
- ▶ « > » : strictement supérieur à ;
- ▶ « <= » : inférieur ou égal à ;
- ▶ « >= » : supérieur ou égal à .

Pour la gestion d'une valeur quelconque de « key », l'opérateur de comparaison prend la valeur « \* ». Dans ce cas, quelle que soit la valeur de provenant de « 1Water » (« key »), la valeur « StaR-DT » prend « value ».

Quand l'opérateur de comparaison vaut « \* », la balise « key » n'est pas prise en compte.

```
"dictionaries": [  
  {  
    "name": "1W_EAU_TRONCON_MODCIRCUL",  
    "matching": [  
      {  
        "key": 1,  
        "value": "enChargeGravite"  
      },  
      {  
        "key": 2,  
        "value": "enChargePressurisation"  
      },  
      {  
        "key": 99,  
        "value": "enChargeGravite"  
      }  
    ]  
  },  
]
```

**Figure 89 : Exemple de dictionnaire avec un opérateur de comparaison d'égalité implicite**

```
{  
  "name": "SUPERIEUR_A_100",  
  "matching": [  
    {  
      "key": 100,  
      "ope": "<=",  
      "value": 0  
    }  
  ]  
}
```

**Figure 90 : Exemple de dictionnaire avec un opérateur de comparaison supérieur ou égal**

La gestion des intervalles est réalisée en utilisant plusieurs correspondances de valeurs successives basées sur des opérateurs de comparaison. La première correspondance trouvée est prise en compte.

### 5.3.5.4 Domains

Cette partie contient un tableau représentant les domaines (« name »), au sens Utility Network, pris en compte au niveau de « 1Water » pour réaliser l'export de données au format « StaR-DT ».

Chaque section contient la description des éléments de « 1Water » à exporter et à retranscrire vers le format « StaR-DT » (« [sections](#) »).

```

"domains": [
  {
    "name": "RES_EAU",
    "sections": {
      "Accessoire": [
        "CanalisationEau": [
        ]
      ]
    }
  },
  {
    "name": "RES_EU",
    "sections": {
      "Accessoire": [
        "CanalisationAssainissementPluviale": [
        ],
        "RegardVisitable": [
        ],
        "RegardNonVisitable": [
        ]
      ]
    }
  },
  {
    "name": "RES_EP",
    "sections": {
      "Accessoire": [
        "CanalisationAssainissementPluviale": [
        ],
        "RegardVisitable": [
        ],
        "RegardNonVisitable": [
        ]
      ]
    }
  }
]

```

Figure 91 : Partie "domains"

### 5.3.5.5 Sections

La partie « sections » décrit la conversion du modèle « 1Water » vers le modèle « StaR-DT » pour chaque section.

! Chaque élément de « sections » doit correspondre à un élément du groupe « headers ».

Une section est représentée par une liste d'entités du modèle « 1Water » au sens Utility Network.

Une entité est définie par :

- ▶ **source** : Line, Junction ou Device ;
- ▶ **group** : code de l'AssetGroup ;
- ▶ **type** : code de l'AssetType (-1 pour tous les AssetType de l'AssetGroup).

```

"source": "Line",
"group": 100,
"type": -1,

```

Figure 92 : Entité d'une section

Une entité est éventuellement associée à un ou plusieurs filtres attributaires (« filters »). Un filtre attributaire est défini par un attribut du modèle « 1Water » (« attribute »), un opérateur de comparaison (« ope ») et une valeur de comparaison (« value »).

```

"filters": [
  {
    "attribute": "visitable",
    "ope": "==",
    "value": 1
  }
],

```

Figure 93 : Filtre sur une entité

Les opérateurs de comparaison possibles sont :

- ▶ « == » : égal à ;
- ▶ « != » : différent de ;
- ▶ « < » : strictement inférieur à ;
- ▶ « > » : strictement supérieur à ;
- ▶ « <= » : inférieur ou égal à ;

- « >= » :supérieur ou égal à.

Une entité est ensuite associée à une liste de définitions de valeur « values » pour renseigner chaque colonne déclarée dans la partie « headers » de la section.

La liste de définitions de valeur est soit déclarée directement dans le groupe « values », soit le groupe « values » est associé à un de [modèle d'attributs « modelatt »](#).

- ! Que la liste de définitions de valeur soit directement dans le groupe « values » ou associée à un modèle, l'ordre des champs (correspondance) doit correspondre à l'ordre attendu par l'élément du groupe « headers ».

Une valeur est représentée par :

- **attribute** : nom de l'attribut « 1Water » ;
- **format** : format texte d'écriture de la valeur de l'attribut au sens Python ;
- **dictionary** : nom du dictionnaire, défini dans la partie « [dictionaries](#) », pour convertir une valeur (optionnel) ;
- **complete** : nom du groupe de complément d'attribut dans la partie « [completeatt](#) » (optionnel) ;
- **formule** : chaîne décrivant une formule utilisable avec la méthode « eval » du langage Python (optionnel) ;
- **ope** : opération de comparaison associée à « value » (optionnel) ;
- **value** : valeur à comparer avec celle de « attribute » en utilisant l'opération « ope » (optionnel) ;
- **next** : permet de définir un autre attribut « 1Water » à lire si « attribute » n'est pas renseigné (optionnel) ou si la condition « attribut » avec « ope » et « value » est vrai.

Dans « next », il est possible de lister plusieurs attributs à lire successivement tant qu'aucune valeur n'est récupérée.

La définition d'un bloc « next », pour chaque attribut, contient :

- **attribute** : nom de l'attribut « 1Water » ;
  - **ope** : opération de comparaison associée à « value » (optionnel) ;
  - **value** : valeur à comparer avec celle de « attribute » en utilisant l'opération « ope » (optionnel) ;
- **default** : valeur à utiliser si « attribute » ou le résultat de « next » ne donne pas de résultat.

```
{  
  "attribute": "diametre",  
  "format": "%0.2f",  
  "dictionary": "SUPERIEUR_A_100",  
  "next": [  
    {  
      "attribute": "diamint"  
    },  
    {  
      "attribute": "diamext"  
    }  
  ],  
  "default": ""  
},
```

Figure 94 : Valeur avec dictionnaire et « next »

### 5.3.5.6 Attributs

Cette partie permet de déclarer des modèles d'informations attributaires partagés par plusieurs sections des « domains ».

#### 5.3.5.6.1 Groupe « modelatt »

La déclaration d'un modèle d'attributs « modelatt » permet d'éviter de déclarer plusieurs fois la description de valeurs d'attribut si elle est partagée par plusieurs sections.

Un modèle d'attributs est défini par un nom (« name ») et associé à une liste de définitions de valeur (« values »). Le nom d'un modèle d'attributs est associé à une balise « values » d'une section.

La description de la liste de définitions de valeur est la même que celle [décrite dans la « sections »](#).

```
"modelatt": [
  {
    "name": "att_Accessoire_Device_RES_EAU",
    "values": [
      {
        "attribute": "id",
        "format": "%ld",
        "dictionary": "",
        "next": [],
        "default": 0
      },
      {
        "attribute": "ident",
        "format": "%s",
        "dictionary": "",
        "next": [],
        "default": ""
      },
      {
        "attribute": "qualitegeolocxy",
        "format": "%s",
        "dictionary": "1W_CLASSE_PRECISION",
        "next": [],
        "default": 3
      },
      {
        "attribute": "qualitegeolocx",
        "format": "%s",
        "dictionary": "1W_CLASSE_PRECISION",
        "next": [],
        "default": 3
      }
    ]
  }
],
```

**Figure 95 : Exemple d'un modèle d'attributs**

#### 5.3.5.6.2 Paramètres

Dans la description d'une définition de valeur, la balise « format » permet d'indiquer le format d'écriture d'une valeur dans le fichier résultat « StaR-DT ».

Dans certains cas, il peut être intéressant de récupérer une valeur sans pour autant l'écrire dans le résultat. Si la balise « format » n'est pas renseignée, la valeur est conservée en mémoire sous le nom de l'étiquette « attribute » pour pouvoir l'utiliser par la suite.

L'exemple ci-dessous indique comment créer une valeur de type texte correspondant à la concaténation du champ « assetgroup », du caractère « / » et du champ « assettype ».

```
{
  "attribute": "assetgroup",
  "format": "",
  "dictionary": "",
  "next": [],
  "default": ""
},
{
  "attribute": "assettype",
  "format": "",
  "dictionary": "",
  "next": [],
  "default": ""
},
{
  "formule": "str(assetgroup) + '/' + str(assettype)",
  "attribute": "",
  "format": "%s",
  "dictionary": "1W_TYPE_ACCESS_DEVICE_RES_EAU",
  "next": [],
  "default": ""
},
{
  "attribute": "",
  "format": "%d",
  "dictionary": "",
  "next": [],
  "default": 0
},
}
```

**Figure 96 : Concaténation de 2 valeurs**

Puis de chercher sa correspondance dans le dictionnaire « 1W\_TYPE\_ACCESS\_DEVICE\_RES\_EAU » afin de trouver la valeur à stocker dans l'attribut de la table « StaR-DT ».

```
"dictionaries": [
  {
    "name": "1W_TYPE_ACCESS_DEVICE_RES_EAU",
    "matching": [
      {
        "key": "150/0",
        "value": ""
      },
      {
        "key": "150/1",
        "value": "pressureRelieveValve"
      },
      {
        "key": "150/2",
        "value": "airRelieveValve"
      },
      {
        "key": "150/3",
        "value": "waterDischargePoint"
      }
    ]
  }
]
```

**Figure 97 : Rechercher une valeur dans un dictionnaire**

## 5.4 MÉTHODE DE NOTATION

1Water propose un système de notation basé sur le calcul d'une note pour un équipement en s'appuyant sur un ensemble de critères ou sur un « assemblage » de notes calculées.

Les équipements concernés sont, principalement, les conduites :

- ▶ EP\_Tronçon : Eaux Pluviales ;
- ▶ EU\_Tronçon : Eaux Usées ;
- ▶ E-Tronçon : Eau Potable.

### 5.4.1 Note

Une note est définie par :

- ▶ un nom (**name**) ;
- ▶ une description (**description**) ;



- ▶ un module (**module**) : métier auquel doivent appartenir les équipements sur lesquels porte la note ;
  - ▶ une note maximale (**size**) : la note est comprise en 1 et « size » ;
  - ▶ une méthode de calcul (**formula**):
    - STANDARD : calcul d'une note par seuils ;
    - SUM : calcul d'une note par somme ;
    - SUMWEIGHTING : calcul d'une note par somme avec pondération ;
  - ▶ une définition de chacun des critères de la note (**criteria**) ;
  - ▶ un type (**standalone**) :
    - principal (standalone=true) ;
    - secondaire (standalone=false).
- ! Seules les notes principales sont proposées à l'utilisateur. Les notes secondaires sont utilisées comme « critères » d'une note principale.

Pour chaque critère, on calcule un indice, puis un tableau permet de retourner une note entière en fonction de la valeur de l'indice calculé.

Les critères peuvent porter sur des éléments directement liés à l'ouvrage ou bien être une combinaison de notes préexistantes. Dans ce cas, on dit que la note est composée. Dans le modèle actuel on distingue plusieurs types de critères.

Le mécanisme de sous note peut être récursif.

Les notes sont modélisées dans un fichier JSON. Un unique fichier modélise toutes les notes sous forme d'une liste de notes.

#### 5.4.1.1 Formule « STANDARD »

Cette méthode permet de déterminer une note finale en comparant le résultat (SCORE) à des seuils définis.

Les seuils sont définis à partir des informations suivantes :

- ▶ SIZE : Valeur de la note maximale (une caractéristique de la note) ;
- ▶ RAWNOTE (critère) : Note retournée par le critère ;
- ▶ MINRAWNOTE (critère) : Note minimale possible pour le critère ;
- ▶ MAXRAWNOTE (critère) : Note maximale possible pour le critère ;
- ▶ WEIGHING (critère) : Pondération du critère dans le calcul de la note finale (une caractéristique du critère) ;
- ▶ SCORE : Somme des produits de chaque critère avec sa pondération ;
- ▶ SCORE\_MIN : Somme des produits de la note minimale de chaque critère avec sa pondération ;
- ▶ SCORE\_MAX : Somme des produits de la note maximale de chaque critère avec sa pondération ;
- ▶ DELTA : Valeur du « pas » entre le score minimum (SCORE\_MIN) et le score maximum (SCORE\_MAX) en fonction de la note maximale (SIZE).

Partant des informations précédentes, les intervalles sont définis pour chaque note entre 1 et la note maximale (SIZE).

- ! Si, pour un critère, la valeur n'est pas définie et qu'aucune valeur par défaut n'est configurée, le critère n'est pas pris en compte dans le calcul.

#### 5.4.1.2 Formule « SUM »

Cette méthode calcule la note finale à partir de la somme des notes de critères pondérés.

Les informations nécessaires sont :

- ▶ RAWNOTE (critère) : Note retournée par le critère ;
- ▶ WEIGHING (critère) : Pondération du critère dans le calcul de la note finale (une caractéristique du critère).

#### 5.4.1.3 Formule « SUMWEIGHTING »

Cette méthode calcule la note finale à partir de la somme des notes de critères pondérés divisée par la somme des pondérations.

Les informations nécessaires sont :

- ▶ RAWNOTE (critère) : Note retournée par le critère ;
- ▶ WEIGHING (critère) : Pondération du critère dans le calcul de la note finale (une caractéristique du critère).

### 5.4.2 Critère

Chaque critère est défini par un ensemble de caractéristiques communes :

- ▶ un type (**type**) ;
  - ▶ un nom (**name**) ;
  - ▶ une description (**description**) (optionnelle) ;
  - ▶ un poids (**weighting**) : poids du critère dans le calcul de la note ;
  - ▶ une valeur par défaut (**defaultScore**) (optionnelle) : permet, si elle est définie, de retourner une note même si l'attribut n'est pas renseigné ou si aucune correspondance n'est trouvée.
- ! Si un critère ne retourne pas de note, il n'est pas pris en compte dans le calcul de la note principale ou secondaire.

En fonction du type de critère, des caractéristiques complémentaires viennent s'ajouter.

#### 5.4.2.1 Critère Attribut Simple

Ce critère permet de calculer un indice en fonction de la valeur d'un attribut de l'équipement. Il est possible d'appliquer une transformation à la valeur de l'attribut avant de calculer la note finale.

Le critère est défini par :

- ▶ **type = SimpleAttribute**
- ▶ un nom (**name**)
- ▶ une description (**description**) (optionnelle)
- ▶ une note par défaut (**defaultScore**) (optionnelle)
- ▶ un nom d'attribut à lire (**attribute**)
- ▶ un nom de transformation à appliquer à la valeur de l'attribut (**transformation**). Cette transformation peut être « null », et dans ce cas aucune transformation ne sera appliquée, ou bien être définie avec l'une des valeurs suivantes :
  - GETAGE : permet pour un attribut de type date de rendre l'âge en nombre d'année entière.
  - GETYEAR : permet pour un attribut de type date de rendre l'année de la date.

- un tableau de conversion permettant de déduire la note du critère à partir de l'indice calculé (**array**).

Principe de calcul :

1. VALUE = valeur de l'attribut **attribute** de l'équipement.
2. Si transformation est null Alors INDICE = VALUE.
3. Si transformation correspond à une fonction prédéfinie F Alors INDICE = F(VALUE).
4. Si transformation ne correspond à aucune fonction prédéfinie Alors INDICE = VALUE.
5. Un tableau permet de définir la note ne fonction de l'indice. Ce tableau peut définir soit des intervalles de valeurs pour l'indice soit une valeur fixe pour l'indice. Chaque ligne du tableau peut indifféremment définir un intervalle ou une valeur fixe pour l'indice. Les lignes du tableau sont parcourues dans l'ordre et la première qui satisfait est sélectionnée.

Exemple d'un critère « Attribut Simple » sans transformation et une comparaison directe de valeur :

```
{
  "type": "SimpleAttribute",
  "name": "NLITPOSE",
  "attribute": "Litpose",
  "weighting": 1.0,
  "array": [
    {
      "value": 0,
      "score": 3
    },
    {
      "value": 1,
      "score": 1
    },
    {
      "value": 2,
      "score": 2
    }
  ]
}
```

Exemple d'un critère « Attribut Simple » avec une transformation et une comparaison par intervalle :

```
{
  "type": "SimpleAttribute",
  "name": "NDATEPOSE",
  "attribute": "datepose",
  "transformation": "GETAGE",
  "weighting": 1.0,
  "array": [
    {
      "minValue": 0,
      "maxValue": 10,
      "score": 0
    },
    {
      "minValue": 10,
      "maxValue": 20,
      "score": 1
    },
    {
      "minValue": 20,
      "maxValue": 30,
      "score": 2
    },
    {
      "minValue": 30,
      "maxValue": 40,
      "score": 3
    },
    {
      "minValue": 40,
      "maxValue": 50,
      "score": 4
    },
    {
      "minValue": 50,
      "maxValue": 1000,
      "score": 5
    }
  ]
},
```

#### 5.4.2.2 Critère Attribut variable

Ce critère permet de calculer une note en fonction de la valeur d'un attribut dont le nom est choisi en fonction de la valeur d'un autre attribut (primaryAttribute). Le choix de l'attribut secondaire est réalisé au moyen d'un tableau qui en fonction de la valeur retourné par l'attribut primaire permet de donner le nom de l'attribut secondaire.

Afin de simplifier ce tableau, on peut désigner une valeur par défaut pour le nom de l'attribut secondaire, et dans ce cas, seules les valeurs de l'attribut primaire qui désignent un attribut secondaire différents de l'attribut par défaut seront présents dans le tableau de conversion. Un fois l'attribut secondaire déterminé, on récupère la valeur et on lui applique une éventuelle transformation pour retrouver l'indice. Puis, par un mécanisme similaire à celui du critère attribut simple on trouve la note en fonction de l'indice.

Exemple : En eau potable, en fonction du matériau, le diamètre à prendre en compte est soit le diamètre nominal soit le diamètre extérieur. Diamètre nominal et diamètre extérieur sont deux attributs différents.

Le critère est défini par :

- ▶ **type = VariableAttribute ;**
- ▶ un nom (**name**) ;
- ▶ une description (**description**) (optionnelle) ;
- ▶ une note par défaut (**defaultScore**) (optionnelle) ;
- ▶ un nom de l'attribut primaire (**primaryAttribute**) permettant de sélectionner l'attribut secondaire ;
- ▶ une valeur par défaut de l'attribut secondaire (**defaultAttribute**) ;
- ▶ une table de conversion qui permet de sélectionner l'attribut secondaire (**attributeArray**) ;
- ▶ un nom d'une transformation à appliquer à la valeur de l'attribut (**transformation**) ;
- ▶ un tableau de conversion permettant de déduire la note du critère à partir de l'indice calculé (**array**).

Principe de calcul :

1. VALUE0 = valeur de l'attribut **primaryAttribute** de l'équipement
2. ATT2 = valeur de l'attribut secondaire trouvé en fonction de VALUE0
3. VALUE = valeur de l'attribut **ATT2** de l'équipement
4. Si transformation est null Alors INDICE = VALUE
5. Si transformation correspond à une fonction prédéfinie F Alors INDICE = F(VALUE)
6. Si transformation ne correspond à aucune fonction prédéfinie Alors INDICE = VALUE
7. Un tableau permet de définir la note en fonction de l'indice. Ce tableau peut définir soit des intervalles de valeur pour l'indice soit une valeur fixe pour l'indice. Chaque ligne du tableau peut indifféremment définir un intervalle ou une valeur fixe pour l'indice. Les lignes du tableau sont parcourues dans l'ordre et la première qui satisfait est sélectionnée.

Exemple d'extrait du JSON qui code un critère « Attribut variable » :

```
{
  "type": "VariableAttribute",
  "name": "NDIAMETRE",
  "primaryAttribute": "matériau",
  "defaultAttribute": "diametre",
  "attributeArray": [
    {
      "value": 16,
      "attribute": "diamext"
    },
    {
      "value": 17,
      "attribute": "diamext"
    },
    {
      "value": 18,
      "attribute": "diamext"
    },
    {
      "value": 24,
      "attribute": "diamext"
    },
    {
      "value": 25,
      "attribute": "diamext"
    }
  ],
  "weighting": 1.0,
  "array": [
    {
      "minValue": 0,
      "maxValue": 150,
      "score": 0
    },
    {
      "minValue": 150,
      "maxValue": 250,
      "score": 1
    },
    {
      "minValue": 250,
      "maxValue": 300,
      "score": 2
    },
    {
      "minValue": 300,
      "maxValue": 400,
      "score": 3
    },
    {
      "minValue": 400,
      "maxValue": 500,
      "score": 4
    },
    {
      "minValue": 500,
      "maxValue": 10000,
      "score": 5
    }
  ]
},
```

### 5.4.2.3 Critère Attribut Multiple

Ce critère permet de calculer une note en fonction de la valeur d'un ensemble d'attributs de l'équipement.

Pour chaque attribut défini, on calcule un indice éventuellement transformé. Puis, on détermine la note du critère en recherchant la première ligne du tableau qui répond à la valeur de chaque attribut. La valeur d'un attribut peut correspondre à une valeur « fixe » ou à un intervalle dans le tableau.

Pour chaque ligne du tableau, chaque colonne peut soit définir une valeur fixe soit un intervalle soit être nulle, dans ce cas, cela signifie que la valeur de l'indice correspondant peut prendre n'importe quelle valeur.

Le critère est défini par :

- ▶ **type = MultiAttributes ;**
- ▶ un nom (**name**) ;
- ▶ une description (**description**) (optionnelle) ;
- ▶ une note par défaut (**defaultScore**) (optionnelle) ;
- ▶ une liste des attributs à lire (**attributes**) ;
- ▶ une liste des transformations (**transformations**) à appliquer à chaque valeur d'attribut retournés. La liste peut soit être nulle, dans ce cas aucune transformation ne sera appliquée, soit être une liste de même cardinalité que la liste des attributs. Une valeur null ou vide dans la liste signifie que la valeur de l'attribut correspondant ne sera pas transformée. Les indices de **attributes** et **transformations** doivent correspondre ;
- ▶ un tableau de conversion permettant de déduire la note du critère à partir de l'indice calculé (**array**).

Principe de calcul :

1. Pour chaque **attributes(i)**
  - a.  $VALUE(i)$  = valeur de l'attribut **attribute(i)** de l'équipement
  - b. Si **transformations** est null Alors  $INDICE(i) = VALUE(i)$
  - c. Si **transformations(i)** correspond à une fonction prédéfinie F Alors  $INDICE(i) = F(VALUE(i))$
  - d. Si **transformations(i)** ne correspond à aucune fonction prédéfinie ou si **transformations(i)** = null ou vide Alors  $INDICE(i) = VALUE(i)$
2. Un tableau permet de définir la note en fonction des indices  $INDICE(i)$ . Ce tableau peut définir soit des intervalles de valeurs pour l'indice, soit une valeur fixe pour l'indice, soit nulle, ce qui correspond à une valeur quelconque de l'indice correspondante. Chaque ligne du tableau peut indifféremment définir un intervalle ou une valeur fixe pour l'indice. Les lignes du tableau sont parcourues dans l'ordre et la première qui satisfait est sélectionnée.

Exemple de code pour un critère « Attribut multiple » :

```
{
  "type": "MultiAttributes",
  "name": "NMATERIAU",
  "attributes": [
    "materiau",
    "datepose"
  ],
  "transformations": [
    "",
    "GETYEAR"
  ],
  "weighting": 1.0,
  "defaultScore": 2,
  "array": [
    {
      "columns": [
        {
          "value": 0
        },
        null
      ],
      "score": 3
    },
    {
      "columns": [
        {
          "value": 1
        },
        null
      ],
      "score": 2
    },
    ...
    {
      "columns": [
        {
          "value": 16
        },
        {
          "minValue": 0,
          "maxValue": 1968
        }
      ],
      "score": 3
    },
    {
      "columns": [
        {
          "value": 16
        },
        {
          "minValue": 1968,
          "maxValue": 10000
        }
      ],
      "score": 2
    }
  ]
}
```



#### 5.4.2.4 Critère anomalie

Ce critère permet d'attribuer une note à l'équipement en fonction des anomalies détectées.

Un paramétrage permet de choisir les anomalies dont on tiendra compte. Par ailleurs il faut aussi sélectionner les anomalies en fonction de leur ancienneté.

Le critère anomalie est défini par :

- ▶ **type = Anomaly ;**
- ▶ un nom (**name**) ;
- ▶ une description (**description**) (optionnelle) ;
- ▶ une liste des types d'anomalies (**anomalyTypes**) à prendre en compte ;
- ▶ un âge par défaut (**defaultAge**) : permet de définir la fenêtre de temps de prise en compte des anomalies 5 par défaut ;
- ▶ un booléen (**rate**) qui détermine le type de critère :
  - **True** : taux de casse par km (taux d'anomalie d'un certain type) sur une période (valeur par défaut) ;
  - **False** : nombre de casse (nombre d'anomalie d'un certain type) sur une période.
- ▶ Un tableau permettant de calculer la valeur de la note en fonction d'un intervalle auquel appartient l'indice.

Principe du calcul :

1. DATE\_REF = (année courante – **defaultAge**)
2. Si date de pose (DATE\_POSE) n'existe pas pour l'équipement alors DATE\_INF = DATE\_REF
3. Si Date de pose existe pour l'équipement Alors
  - e. Si DATE\_REF < DATE\_POSE alors DATE\_INF = DATE\_POSE
  - f. Si DATE\_REF >= DATE\_POSE alors DATE\_INF = DATE\_REF
4. DELTA = année courante – DATE\_INF en nombre d'année entière
5. Si **rate** = Vrai Alors QUOTIENT = DELTA / longueur de l'équipement en km
6. Si **rate** = Faux Alors QUOTIENT = 1
7. N = Nombre d'anomalies porté par l'équipement dont le type est dans la liste des types d'anomalies sélectionnés et qui sont plus récentes que DATE\_INF
8. INDICE = N / QUOTIENT
9. Un tableau permet de définir la note en fonction de l'indice.

Par défaut :

**Tableau 1 : Taux de casse (rate = Vrai)**

Indice	Note
0 <= indice < 0.1	0
0.1 <= indice < 0.15	1
0.15 <= indice < 0.2	2
0.2 <= indice < 0.35	3
0.35 <= indice	4

**Tableau 2 : Nombre de casse (rate = Faux)**

Nb casses	Note
0	1
0 < Nb <= 2	2
2 < Nb <= 4	3
4 < Nb	4

Exemple de code pour un critère « Anomalie » de type « Taux de casse » :

```
{
  "type": "Anomaly",
  "anomalyTypes": [
    1,
    2
  ],
  "defaultAge": 12,
  "name": "TAUX_CASSE",
  "weighting": 1.0,
  "array": [
    {
      "minValue": 0,
      "maxValue": 0.1,
      "score": 0
    },
    {
      "minValue": 0.1,
      "maxValue": 0.15,
      "score": 1
    },
    {
      "minValue": 0.15,
      "maxValue": 0.2,
      "score": 2
    },
    {
      "minValue": 0.2,
      "maxValue": 0.35,
      "score": 3
    },
    {
      "minValue": 0.35,
      "maxValue": 1000,
      "score": 4
    }
  ]
}
```

#### 5.4.2.5 Critère Zone Géographique incluse

Ce critère permet d'attribuer une note à l'équipement s'il intersecte une zone géographique.

! La table utilisée pour modéliser les zones géographiques intervenant dans ce critère est fixe : « Zone de notation (600) ».

Chaque zone géographique de ce type est caractérisée par :

- un type correspondant à un attribut (**type**) ;
- une liste des modules pour lesquels la zone est accessible.

Même si le nom de l'attribut type est fixe, le critère permet de le modifier pour des évolutions futures.

Dans ce critère, si l'équipement intersecte la zone et si la surface de l'intersection dépasse un seuil paramétrable, alors le score prend une valeur fixe paramétrable (**includedScore**). En revanche, si la surface de l'intersection est en deçà du seuil alors le score prend une valeur fixe paramétrable (**notIncludedScore**). Si plusieurs zones contiennent l'équipement, celle fournissant le score le plus élevé sera retenue.

Le critère est défini par :

- **type = GeoZoneIncluded** ;

- ▶ son nom (**name**) ;
- ▶ sa description (**description**) (optionnelle) ;
- ▶ le type de zone à prendre en compte (**zoneType**) ;
- ▶ le pourcentage de la surface de l'équipement inclus dans la zone en deçà duquel il faut diminuer la note finale de 1 (**threshold**) (par défaut 50%) ;
- ▶ la note si inclus (**includedScore**) ;
- ▶ la note si non inclus (**notIncludedScore**).

Principe de calcul :

1. On récupère la liste des zones géographiques utilisables, c'est-à-dire les zones géographiques filtrées par le module auquel appartient la note, et le type de la zone paramétré dans le critère.
2. Si aucune zone existe la note ne sera pas prise en compte.
3. L'équipement a une géométrie linéaire, afin de pouvoir avoir une mesure sur l'intersection de l'équipement avec la zone géographique, on calcule un buffer autour de la géométrie de l'équipement. La taille de ce buffer correspond au diamètre de la canalisation.
4. SEQU = surface du polygone défini à partir de la géométrie linéaire de l'équipement et du buffer défini précédemment.
5. Pour chaque zone géographique sélectionnée :
  - g. On calcule l'intersection entre l'équipement et la zone géographique,
  - h. On calcule SINTER = surface de l'intersection entre l'équipement et la zone géographique
  - i. On calcule le pourcentage d'inclusion :
    - i. SI  $0 < (SINTER / SEQU) < \text{threshold}$  Alors SCORE = **notIncludedScore**
    - ii. SI  $\text{threshold} \leq (SINTER / SEQU)$  Alors SCORE = **IncludedScore**
    - iii. Dans les autres cas la zone n'est prise en compte
6. On garde le score le plus élevé.

Exemple d'extrait du JSON qui code un critère « Zone géographique d'inclusion » :

```
{
  "type": "GeoZoneIncluded",
  "name": "NTRAVAUX",
  "zoneType": 2,
  "notIncludedScore": 2,
  "includedScore": 3,
  "weighting": 1.0
},
```

#### 5.4.2.6 Critère Zone Géographique incluse par niveau

Ce critère permet d'attribuer une note à l'équipement s'il intersecte une zone géographique.

! La table utilisée pour modéliser les zones géographiques intervenant dans ce critère est fixe « Zone de notation par niveau(601) ».

Chaque zone géographique de ce type est caractérisée par :

- ▶ un type correspondant à un attribut (**type**) ;
- ▶ un niveau attribut (**level**) ;
- ▶ une liste des modules pour lesquels la zone est accessible.

Même si le nom des attributs **type** et **level** est fixe, le critère permet de les modifier pour des évolutions futures.

Si l'équipement est inclus dans plusieurs zones géographiques, la plus défavorable est retenue (niveau le plus élevé). Comme pour les critères attributaires, un tableau permet de déduire la note à partir de l'indice. De plus, si l'intersection de la géométrie de l'équipement et de la géométrie de la zone est inférieure à un seuil paramétrable (par défaut 50%) alors la note est diminuée de 1, tant qu'elle reste dans l'intervalle de notes (1 à 4).

Le critère est défini par :

- ▶ **type = GeoZoneLevel ;**
- ▶ son nom (**name**) ;
- ▶ sa description (**description**) (optionnelle) ;
- ▶ le type de zone à prendre en compte (**zoneType**) ;
- ▶ la note par défaut (**defaultScore**) (optionnelle) ;
- ▶ le pourcentage de la surface de l'équipement inclus dans la zone en deçà duquel il faut diminuer la note finale de 1 (**threshold**) (par défaut 50%) ;
- ▶ un tableau de conversion permettant de déduire la note du critère à partir de l'indice calculé (**array**).

Principe de calcul :

1. On récupère la liste des zones géographiques utilisables, c'est-à-dire les zones géographiques filtrées par le module auquel appartient la note, et le type de la zone paramétré dans le critère.
2. Si aucune zone n'existe le critère n'est pas retenu.
3. L'équipement a une géométrie linéaire, afin de pouvoir avoir une mesure sur l'intersection de l'équipement avec la zone géographique, on calcule un buffer autour de la géométrie de l'équipement. La taille de ce buffer correspond au diamètre de la canalisation. Si aucun attribut diamètre (**diamètre extérieur, diamètre nominal, largeur**) n'est renseigné, alors la taille par défaut est de 200 mm.
4. SEQU = surface du polygone défini à partir de la géométrie linéaire de l'équipement et du buffer défini précédemment.
5. Pour chaque zone géographique sélectionnée :
  - j. On calcule l'intersection entre l'équipement et la zone géographique,
  - k. Si l'intersection n'est pas vide Alors INDICE = valeur de l'attribut level de la zone
  - l. On calcule le SCORE à partir du tableau de conversion
  - m. On calcule SINTER = surface de l'intersection entre l'équipement et la zone géographique
  - n. SI (SINTER / SEQU) < threshold Alors SCORE est diminué de 1 en restant dans l'intervalle de notes (si le score vaut 1, on ne diminue pas la note).
6. On garde la zone pour laquelle SCORE est le plus élevé

Exemple d'extrait du JSON qui code un critère « Zone géographique incluse par niveau » :

```
{
  "type": "GeoZoneLevel",
  "name": "NTRAFIC",
  "zoneType": 1,
  "weighting": 1.0,
  "array": [
    {
      "value": 1,
      "score": 1
    },
    {
      "value": 2,
      "score": 3
    },
    {
      "value": 3,
      "score": 5
    }
  ]
},
```

#### 5.4.2.7 Critère matrice de notations

Ce critère combine le score de deux autres notes pour en déduire le score de la note. La combinaison est réalisée au moyen d'une matrice qui à chaque couple de score associe un score final. Bien entendu, ce mécanisme est récursif et N1 ou N2 peuvent elle-même avoir un critère de type matrice.

Exemple d'une matrice entre 2 notes :

		SCORE NOTE N1					
SCORE NOTE N2		0	1	2	3	4	
	0	0	1	2	3	4	
	1	1	1	3	4	3	
	2	2	3	3	4	4	
	3	3	4	3	4	4	
	4	4	4	4	4	4	

Le critère est défini par :

- **type = matrix ;**
- un nom (**name**) ;
- une description (**description**) (optionnelle) ;
- une note par défaut (**defaultScore**) (optionnelle) ;
- un nom de la première note (**score1**) ;
- un nom de la seconde note (**score2**) ;
- une matrice (**array**) permettant de calculer le score final à partir du score de score1 et de score2.

Principe de calcul :

1. VALUE1 = score de la note score1.
2. VALUE2 = score de la note score2.

3. Si une des deux notes VALUE1 ou VALUE2 n'est pas défini, le critère ne sera pas pris en compte,
4. On déduit le SCORE à partir du résultat de la matrice.

Exemple d'extrait du JSON qui code un critère Matrice :

```
{
  "type": "matrix",
  "name": "NOTE_MATRICE",
  "score1": "NOTE_1",
  "score2": "NOTE_2",
  "weighting": 1.0,
  "array": [
    {
      "value1": 0,
      "value2": 1,
      "score": 0
    },
    {
      "value1": 0,
      "value2": 1,
      "score": 1
    },
    {
      "value1": 0,
      "value2": 2,
      "score": 2
    },
    {
      "value1": 0,
      "value2": 3,
      "score": 3
    },
    ...
    {
      "value1": 4,
      "value2": 4,
      "score": 4
    }
  ]
}
```

#### 5.4.2.8 Critère Opération de notations

Ce critère, comme le précédent permet de combiner deux notes. En revanche, la combinaison est réalisée au moyen d'une formule de calcul qui porte sur les deux sous notes obtenues.

Le critère est défini par :

- ▶ **type = operation ;**
- ▶ un nom (**name**) ;
- ▶ une description (**description**) (optionnelle) ;
- ▶ une note par défaut (**defaultScore**) (optionnelle) ;
- ▶ un nom de la première note (**score1**) ;
- ▶ un nom de la seconde note (**score2**) ;
- ▶ un nom de l'opération (**operation**) à effectuer sur les scores intermédiaires pour déterminer le résultat :
  - AVERAGE : moyenne entre score1 et score2 ;
  - MULT : produit entre score1 et score2.

- un tableau de conversion permettant de déduire la note du critère à partir du résultat de l'opération (**array**).

Principe de calcul :

1. Si **operation** est null Alors le critère n'est pas pris en compte.
2. Si **operation** ne correspond à aucune fonction prédéfinie Alors le critère n'est pas pris en compte.
3. SCORE1 = score de la note score1.
4. SCORE2 = score de la note score2.
5. Si **operation** correspond à une fonction prédéfinie F Alors INDICE = F(SCORE1, SCORE2).
6. On déduit le SCORE à partir du tableau de conversion.

Exemple d'extrait du JSON qui code un critère « Opération » :

```
{
  "type": "operation",
  "name": "SCORE_RISQUE",
  "score1": "ALEA",
  "score2": "ENJEU",
  "operation": "MULT",
  "weighting": 1.0,
  "array": [
    {
      "minValue": 0,
      "maxValue": 1,
      "score": 0
    },
    {
      "minValue": 1,
      "maxValue": 6,
      "score": 1
    },
    {
      "minValue": 6,
      "maxValue": 9,
      "score": 2
    },
    {
      "minValue": 9,
      "maxValue": 13,
      "score": 3
    },
    {
      "minValue": 13,
      "maxValue": 17,
      "score": 4
    }
  ]
}
```

## 6 ANNEXES

### 6.1 FILTRES DE SÉLECTION PERSONNALISÉS

Les filtres de sélection sont définis, au niveau de chaque module concerné, dans le groupe « **filters** ».

#### 6.1.1 Généralités

Un filtre est défini par les paramètres suivants :

- ▶ **name** : nom du filtre (obligatoire) ;
- ▶ **description** : description du filtre (optionnelle) ;
- ▶ **condition** : condition de sélection contenant un paramètre *networkEntities* ou un paramètre *whereClause* ou une combinaison de ces 2 paramètres (obligatoire).

! Si la condition n'est pas valide, le filtre est rejeté.

Le paramètre *networkEntities* représente la liste des éléments du réseau sur lesquels la condition s'applique. Chaque entrée du paramètre *networkEntities* est définie par : *source*, *assetGroup*, *domainNetworkName* et éventuellement *assetType*.

! Les éléments renseignés dans le paramètre *networkEntities* doivent être présents dans la liste des éléments du réseau gérés par le module concerné. Si ce n'est pas le cas (même pour un seul de ces éléments), le filtre concerné est rejeté.

! Si le *domainNetworkName* d'un élément de *networkEntities* est incorrect ou inconnu, le filtre concerné est rejeté.

! Si l'*assetGroup* d'un élément de *networkEntities* est incorrect ou inconnu pour le domaine et la source indiqués, le filtre concerné est rejeté.

! Si un *assetType* est renseigné mais est incorrect ou inconnu pour l'*assetGroup* indiqué, le filtre concerné est rejeté.

Le paramètre *whereClause* représente la condition à appliquer sur les éléments du réseau. Le contenu de *whereClause* doit respecter le formalisme des requêtes faites aux services REST ArcGIS. Si le paramètre *networkEntities* est absent, le paramètre *whereClause* s'applique sur les éléments du réseau gérés par le module concerné.

#### 6.1.2 Assainissement

```
"filters": [  
  {  
    "name": "Eaux usées",  
    "description": "Equipements de type eaux usées",  
    "condition": {  
      "whereClause": "reseau = 2"  
    }  
  },  
  {  
    "name": "Eaux pluviales",  
    "description": "Equipements de type eaux pluviales",  
    "condition": {  
      "whereClause": "reseau = 1"  
    }  
  }  
]
```



```
},
{
  "name": "Linéaires < 15m",
  "description": "Tronçons et branchement avec une longueur inférieure à 15m",
  "condition": {
    "whereClause": "longueur < 15",
    "networkEntities": [
      {
        "source": "esriUNFCUTLine",
        "assetGroup": 200,
        "domainNetworkName": "RES_ASS"
      },
      {
        "source": "esriUNFCUTLine",
        "assetGroup": 201,
        "domainNetworkName": "RES_ASS"
      },
      {
        "source": "esriUNFCUTLine",
        "assetGroup": 203,
        "domainNetworkName": "RES_ASS"
      },
      {
        "source": "esriUNFCUTLine",
        "assetGroup": 204,
        "domainNetworkName": "RES_ASS"
      }
    ]
  }
},
{
  "name": "Tronçons EU > 40 m",
  "description": "Tronçons de type eaux usées avec une longueur est supérieure à 40m",
  "condition": {
    "whereClause": "reseau = 2 AND longueur > 40",
    "networkEntities": [
      {
        "source": "esriUNFCUTLine",
        "assetGroup": 201,
        "domainNetworkName": "RES_ASS",
        "assetType": null
      }
    ]
  }
},
{
  "name": "Tronçons EU séparatifs",
  "description": "Tronçons de type eaux usées en séparatifs ",
  "condition": {
    "networkEntities": [
      {
        "source": "esriUNFCUTLine",
        "assetGroup": 201,
        "domainNetworkName": "RES_ASS",
        "assetType": 1
      }
    ]
  }
},
]
```

### 6.1.3 Eau

```
"filters": [  
  {  
    "name": "Distribution",  
    "description": "Tronçons de type distribution",  
    "condition": {  
      "networkEntities": [  
        {  
          "source": "esriUNFCUTLine",  
          "assetGroup": 100,  
          "assetType": 2,  
          "domainNetworkName": "RES_EAU"  
        }  
      ]  
    }  
  },  
  {  
    "name": "Production",  
    "description": "Tronçons de type production",  
    "condition": {  
      "networkEntities": [  
        {  
          "source": "esriUNFCUTLine",  
          "assetGroup": 100,  
          "assetType": 1,  
          "domainNetworkName": "RES_EAU"  
        }  
      ]  
    }  
  },  
  {  
    "name": "Transport",  
    "description": "Tronçons de type transport",  
    "condition": {  
      "networkEntities": [  
        {  
          "source": "esriUNFCUTLine",  
          "assetGroup": 100,  
          "assetType": 3,  
          "domainNetworkName": "RES_EAU"  
        }  
      ]  
    }  
  },  
  {  
    "name": "Tronçons < 25m",  
    "description": "Tronçons avec une longueur inférieure à 25m",  
    "condition": {  
      "networkEntities": [  
        {  
          "source": "esriUNFCUTLine",  
          "assetGroup": 100,  
          "assetType": null,  
          "domainNetworkName": "RES_EAU"  
        }  
      ]  
    },  
    "whereClause": "longueur < 25"  
  }  
],
```

```
{
  "name": "Tronçons 2023",
  "description": "Tronçons posés en 2023",
  "condition": {
    "whereClause": "datepose >= TIMESTAMP '2023/01/01 00:00:00' AND datepose <= TIMESTAMP '2023/12/31 23:59:59'",
    "networkEntities": [
      {
        "source": "esriUNFCUTLine",
        "assetGroup": 100,
        "assetType": null,
        "domainNetworkName": "RES_EAU"
      }
    ]
  }
}
```

## 6.2 DÉFINIR LES CARTES

La définition des cartes, au format « JSON », exploitée par 1Network Manager se trouve dans le répertoire des données complémentaires et des paramètres de 1Network Manager « \1Spatial\Maps\[fr ou en] ».

! Les fichiers de définition de carte livrés s'appuient sur le modèle standard 1Water mais l'administrateur peut les redéfinir en fonction de sa modélisation et de ces besoins sous certaines conditions.

La syntaxe des fichiers de définition des cartes contient les balises suivantes :

► **type** : type de carte parmi 9 types possibles :

- **agpMapType** : carte de travail pour le client 1Water Desktop (ArcGIS Pro) ;
- **agpAdvancedMapType** : carte de travail avancée, basée sur une configuration plus détaillée de la construction de la carte, pour le client 1Water Desktop (ArcGIS Pro) ;
- **baseMapType** : carte « vide » utilisée pour l'initialisation des cartes Web avancées (assainissement et eau) au format ArcGIS Pro avec la génération d'un fichier JSON de description contenant uniquement un fond de plan.
- **agpWebMapType** : carte pour ArcGIS Pro contenant tous les éléments nécessaires à la définition d'une carte Web (réseau de distribution, couches d'exploitation et éventuellement couches de sectorisation) pour les clients 1Water Web et 1Water Mobile.  
Cette carte complète une carte existante, par exemple la carte « vide » (baseMapType), publiée au préalable et importée dans le projet ArcGIS Pro.  
La mise à jour de la carte est réalisée à l'aide de l'outil standard d'ArcGIS Pro « Partager - Gérer - Enregistrer la carte Web » ;
- **exploitationMapType** : carte basée sur des couches d'exploitation métier ;
- **simpleMapType** : carte basée sur des couches complémentaires ;
- **unMapType** : carte basée sur des couches d'un réseau de distribution (Utility Network) ;
- **webMapDisplayType** : carte pour les clients 1Water Web et 1Water Mobile ;
- **webMapFormType** : carte des formulaires pour les clients 1Water Web et 1Water Mobile.

► **label** : libellé associé à la carte au niveau de 1Network Manager ;

- ▶ **name** : nom de la carte affiché au niveau du portail ;
- ▶ **description** : résumé de la carte affiché au niveau du portail ;
- ▶ **style** : nom du fichier de styles exploités pour la création de la carte ;
- ▶ **layers** : liste des couches définies dans la carte.

Chaque élément du « tableau » layers peut être défini avec les balises suivantes :

- ▶ **name** : nom de la table ou nom de la couche ou « group » (en minuscules) ;
- ▶ **label** : libellé associé à l'élément dans la description de la carte ;
- ▶ **visibility** : la couche est affichée (true) ou masquée (false). Par défaut, la couche est masquée à l'ouverture de la carte ;
- ▶ **istable** : une table (true) ou une couche (false) ;
- ▶ **layerid** : identifiant de la source, de la table ou de la couche ;
- ▶ **group** : tableau contenant la liste des « layerid » ou des « AssetGroup » regroupés dans cet ensemble ;
- ▶ **uniquevaluefield** : nom du champ de la couche utilisé par le fichier de styles pour définir la symbologie en fonction de la valeur pour les couches d'exploitation et d'habillage ;
- ▶ **symbol** : catégorie du symbole, présent dans le fichiers de style à utiliser pour la couche (facultatif avec, par défaut, le nom de la table de stockage) ;
- ▶ **minscale** : échelle minimum d'affichage de la couche ;
- ▶ **maxscale** : échelle maximum d'affichage de la couche ;
- ▶ **popup** : utilisé par 1Water Mobile pour la description d'un élément dans la liste d'information. Le popup permet de définir les balises suivantes :
  - **expressions** : liste d'expressions Arcade à utiliser dans les lignes du popup. Chaque expression est définie par :
    - **name** : nom interne de l'expression ;
    - **title** : libellé utilisateur dans les dialogues de configuration ;
    - **expression** : script Arcade ;
  - **description** : information sur le popup ;
  - **rows** : liste de chaînes de caractères définissant chacune une ligne dans le popup ;
- ▶ **formulaire** : utilisé par 1Water Mobile pour décrire le formulaire de consultation et de modification d'un élément d'exploitation. Cette partie s'appuie sur le principe de Field Maps. Le formulaire permet de définir les balises suivantes :
  - **title** : titre du formulaire ;
  - **expressions** : liste d'expressions Arcade (même définition que pour le popup ci-dessus) ;
  - **groups** : liste des groupes du formulaire. Chaque groupe est défini par les balises suivantes :
    - **title** : titre du groupe ;
    - **collapsed** : flag indiquant si le groupe est replié ou déplié à l'ouverture du formulaire ;
    - **visible** : Flag indiquant si le groupe est visible dans le formulaire ;
    - **fields** : liste des champs à présenter dans le groupe. Chaque champ est défini par les balises suivantes :
      - **fieldname** : nom du champ ou de l'expression à afficher ;
      - **required** : flag indiquant si la saisie du champ est obligatoire ;
      - **editable** : flag indiquant si le champ peut être modifié ;
      - **visibilityExpression** : expression Arcade conditionnant la visibilité du champ dans le formulaire ;

- **includeTime** : pour un champ de type DATE, flag indiquant s'il faut ou non afficher l'heure ;
- **subtypes** : liste de déclaration des sous-types d'une couche. Permet de définir, pour chaque « Asset Group » du layer courant, les propriétés suivantes :
  - **code** : identifiant numérique de l'Asset Group ;
  - **label** : Libellé à utiliser (facultatif) ;
  - **render** : champ utilisé pour un rendu par valeur unique (si non défini, rendu simple) ;
  - **visibility** : flag indiquant la visibilité par défaut ;
  - **minscale** : échelle d'affichage minimum ;
  - **maxscale** : échelle d'affichage maximum ;
  - **symbol** : catégorie du symbole dans le fichier de styles à utiliser ;
  - **etiquettes** : flag pour indiquer la mise en place d'étiquettes de connectivité basées sur les noms de port « upstream » de configuration de terminal pour les éléments linéaires du réseau (non géré pour les cartes Web) ;
  - **orientation** : définition de la règle d'orientation pour un symbole ponctuel ;
  - **coloroverride** : définition d'une règle de surcharge de la couleur pour les symboles ponctuels (non géré pour les cartes Web) ;

Les informations ci-dessus précisent l'usage des balises en fonction du type de carte :

! L'usage des balises varie en fonction du type de carte choisi.

- **agpMapType** : définition de la carte ArcGIS Pro pour le client 1Water.
  - Source de données : réseau de distribution au travers d'un service d'entités ;
  - Le fichier de styles et la définition du réseau de distribution fournissent les informations nécessaires pour l'organisation de la carte. Tous les autres paramètres des couches sont définis dans le code de l'application ;
  - Seules les couches complémentaires au réseau de distribution doivent être décrites dans le JSON (ex : fonde de plan par défaut et table des logs d'opération);
- **agpAdvancedMapType** : définition de la carte ArcGIS Pro pour le client 1Water. Tous les paramètres définis pour la configuration des cartes Web sont utilisés pour la définition de cette carte.
  - Source de données : réseau de distribution au travers d'un service d'entités ;
  - Le fichier de styles, la définition du réseau de distribution et les caractéristiques au niveau de chaque layer fournissent les informations nécessaires pour l'organisation de la carte ;
  - L'attribut « **name** » caractérise certains éléments particuliers de la définition à l'aide des mots clés suivants :
    - **basemap** : caractéristiques de la couche de fond de plan ;
    - **log\_operation** : caractéristique de la couche de l'historisation des opérations réalisées ;
  - lien entre la définition des sources du réseau et les identifiants de layer avec le nom du réseau et le type d'usage des sources comme par exemple :
    - **networkRES\_EAU** : groupe de couches pour le domaine de réseau « RES\_EAU » ;
    - **StructureLine** : source de données linéaire du réseau de structure ;

- LineRES\_EAU : source de données linéaires du domaine de réseau « RES\_EAU » ;
    - L'attribut « **layerid** » indique l'identifiant d'une source de donnée. L'utilisation d'identifiant négatif permet de caractériser les groupes de couches pour les organiser entre eux ;
- ! Ce modèle de carte est très fortement lié à la définition du réseau.
- **baseMapType** : carte d'initialisation pour les cartes Web avancées avec uniquement les caractéristiques de la couche de fond de plan « **basemap** » au niveau du groupe « layers ».
  - **agpWebMapType** : carte ArcGIS Pro servant de base à la définition d'une carte Web pour un usage avec les clients 1Water Web et 1Water Mobile :
    - Source de données : réseau de distribution par le service d'entités du patrimoine, service d'exploitation et service de sectorisation ;
    - Le fichier de styles, la définition du réseau de distribution et les caractéristiques au niveau de chaque layer fournissent les informations nécessaires pour l'organisation de la carte ;
    - Configuration similaire à la carte « **agpAdvancedMapType** » pour la partie patrimoine ;
    - Configuration similaire à la carte « **webMapDisplayType** » pour les sources d'exploitation et de sectorisation ;
    - Prise en compte de la balise « **subtypes** » permet de définir plus en détail les différents groupes d'éléments du réseau (voir les attributs disponibles).
  - **simpleMapType** : carte simple pour une publication en tant que service d'entités (ex : carte de sectorisation) :
    - L'utilisateur définit une source de données (fichier de connexion vers la base de données d'entreprise). Seules les **couches de polygones** sont prises en compte dans cette carte ;
    - Le fichier de styles et la définition des couches fournissent les informations nécessaires pour l'organisation de la carte ;
    - L'attribut « **name** » indique le nom court de la couche à utiliser dans la source de données fournie ;
    - L'attribut « **layerid** » indique l'identifiant de layer à affecter à cette source dans la carte ;
    - Les autres attributs d'un layer pris en compte : « **label** », « **visibility** », « **uniquevaluefield** », « **symbolcategorie** », « **minscale** » et « **maxscale** ».
  - **exploitationMapType** : carte de publication des données d'exploitation :
    - L'utilisateur définit une source de données (fichier de connexion vers la base de données d'entreprise) ;
    - Le fichier de styles et la définition des couches fournissent les informations nécessaires pour l'organisation de la carte ;
    - L'attribut « **name** » indique le nom court de la couche ou de la table à utiliser dans la source de données fournie ;
    - Fonctionnement similaire à la carte « **simpleMapType** » pour les couches (ponctuel, linéaire ou surfacique) et tables (attribut « **istable** » pris en compte) d'exploitation.
  - **unMapType** : carte de publication pour un réseau de distribution :
    - L'utilisateur définit un réseau de distribution comme source de donnée par l'intermédiaire d'une connexion vers la base de données (connexion en mode versionnement de branche obligatoire) ;

- Le fichier de styles et la définition du réseau de distribution fournissent les informations nécessaires pour l'organisation de la carte. Tous les autres paramètres des couches sont définis dans le code de l'application ;
  - Seules les couches et tables complémentaires, n'appartenant pas au réseau et présentent dans le même espace de données, sont à lister avec le même principe de définition que la carte « exploitationMapType » (ex : couches de validation des erreurs et table des logs d'opération).
- **webMapDisplayType** : définition de carte Web au format JSON. Héritage de la définition des services d'entités notamment pour la gestion des symboles :
- Source de données : service d'entités du patrimoine (réseau de distribution) et d'exploitation ;
  - Le fichier de styles, la définition du réseau de distribution et les caractéristiques au niveau de chaque layer fournissent les informations nécessaires pour l'organisation de la carte ;
  - L'attribut « **name** » caractérise certains éléments particuliers de la définition à l'aide des mots clés suivants :
    - **exclude** : exclusion d'une liste de layerid définis dans l'attribut « **group** » ;
    - **basemap** : caractéristiques de la couche de fond de plan ;
    - **group** : définition d'un groupe de couches pour organiser les couches dans la carte ;
  - L'attribut « **layerid** » indique l'identifiant d'une source de donnée. L'utilisation d'identifiant négatif permet de caractériser les groupes de couches pour les organiser entre eux ;
  - Les attributs « **popup** » et « **formulaire** » peuvent être définis pour les couches d'exploitations et sont utilisés pour la définition de la carte de formulaires. Pour les éléments du patrimoine, la définition des popup et des formulaires provient de la configuration du réseau associé aux données.
- **webMapFormType** : définition de carte Web pour la gestion des popup et des formulaires :
- Source de données : service d'entités du patrimoine (réseau de distribution) et d'exploitation ;
  - L'attribut « **name** » caractérise certains éléments particuliers de la définition à l'aide des mots clés suivants :
    - **exclude** : exclusion d'une liste de layerid définis dans l'attribut « **group** » ;
    - **basemap** : caractéristiques de la couche de fond de plan ;
    - **group** : définition d'un groupe de couches pour organiser les couches dans la carte ;
  - L'attribut « **layerid** » indique l'identifiant d'une source de donnée. L'utilisation d'identifiant négatif permet de caractériser les groupes de couches pour les organiser entre eux ;
  - Les attributs « **popup** » et « **formulaire** » peuvent être définis pour les couches d'exploitations et sont utilisés pour la définition de la carte de formulaires. Pour les éléments du patrimoine, la définition des popup et des formulaires provient de la configuration du réseau associé aux données.

## 6.3 GESTION DE LA VISIBILITÉ DES ASSOCIATIONS DE CONTENANCE

La mise en place de la configuration de la visibilité des associations de contenance, au niveau des cartes ArcGIS Pro et Web, est nécessaire seulement si la modélisation contient une définition de structure ou d'assemblage.

Pour chaque métier et chaque carte concernés, les couches suivantes doivent être configurées :

- ▶ Linéaire ;
- ▶ Jonction ;
- ▶ Appareil ;
- ▶ Assemblage.

Pour configurer le filtre d’affichage sur ces couches, reportez-vous à la documentation Esri sur le [contrôle de la visibilité des associations](#) au niveau du paragraphe « Configurer les filtres d’affichage à l’aide de l’attribut d’état de l’association ».

## 6.4 PROFIL EN LONG

La configuration du module « Profil en long », de chaque application, est définie dans le fichier « `\config\onewater\<application>\longprofile.json` ».

Le mot clé « `<application>` » peut prendre la valeur « water » ou « sanitation ».

! Si une option est facultative, l’information est précisée dans la description suivante. Sinon, la balise doit obligatoirement être indiquée dans le fichier « longprofile.json ».

### 6.4.1 Configuration de la trace

Un profil en long s’appuie sur une trace de « chemin le plus court » entre deux ponctuels du patrimoine d’un « réseau de distribution » (Utility Network).

La configuration s’appuie sur une trace nommée qui exploite des catégories spécifiques (LONG\_PROFIL\_LINEAR, LONG\_PROFIL\_NODE et LONG\_PROFIL\_CONNECTION) pour préciser le résultat du profil en long.

La configuration de la trace est définie par les clés « **traceConfigurationName** » et « **startingPointCategoryNames** ».

```
"traceConfigurationName": "1W_WATER_LONG_PROFIL",  
"startingPointCategoryNames": [  
    "LONG_PROFIL_LINEAR"  
],  
"startingPointGeometryType": "node",  
"canAddBarriers": false,
```

Figure 98 : Configuration de la trace pour le profil en long

- ▶ **traceConfigurationName** : nom de la trace utilisée pour calculer le profil en long ;
- ▶ **startingPointCategoryNames** : liste des catégories de type linéaire pour sélectionner le point de départ et le point d’arrivée de la trace de chemin le plus court. A partir d’une catégorie de linéaire, récupération des jonctions et des appareils qui peuvent être connectés.
- ▶ **startingPointGeometryType** : indique le type de géométrie du point de départ et arrivée (uniquement « **node** » dans cette version).
- ▶ **canAddBarriers** : indique si on peut ajouter des barrières (true ou false). Une barrière permet de bloquer un chemin entre le point de départ et le point d’arrivée.

### 6.4.2 Configuration du rapport

La configuration du rapport est définie dans le groupe « `"report": {` ».



### 6.4.2.1 Paramètres généraux

```
"defaultUnity": "m",
"dateFormat": "dd-MM-yyyy",
"logo": {
  "name": "images/1Water.png",
  "width": 150,
  "height": 40
},
```

Figure 99 : Paramètres généraux du profil en long

- **defaultUnity** : unité par défaut à utiliser dans les « **fieldExpressionsByPriority** ». Les valeurs possibles sont :
  - **m** : mètre (préconisé par rapport à l'unité des données) ;
  - **cm** : centimètre ;
  - **mm** : millimètre.
- **dateFormat** : format de la date du jour. Les formats supportés sont :
  - **dd/MM/yyyy** (ex : 01/01/2025) ;
  - **dd-MM-yyyy** (ex : 01-01-2025).
- **logo** : logo à mettre sur le profil en long ;
  - **name** : chemin relatif du fichier à partir du répertoire « **\config\onewater\<application>** » avec « **<application>** » pouvant prendre la valeur « **water** » ou « **sanitation** » ;
  - **width** : largeur en pixel;
  - **height** : hauteur en pixel.

### 6.4.2.2 Tableau d'informations

Le tableau d'information d'un profil en long est affiché sous le tracé du profil :

Longueur (m)	38.30	48.84	54.84
Pente en mm/m	48.30	49.83	39.40
Hauteur/largeur (mm)	200/200	200/200	200/200
Matériau	Béton gris	Béton gris	Béton gris

Figure 100 : Tableau d'information du profil en long

Cette zone est définie par la section « **bottomTable** » dans le fichier « **longProfile.json** » :

```
"bottomTable": {
  "freeFields": [
```

Figure 101 : Définition du tableau d'informations du profil en long

- **bottomTable** : tableau des caractéristiques des tronçons du profil.
- ! Les 2 premières lignes du tableau contiennent la longueur et la pente de chaque tronçon (comportement par défaut, non configurable)

- **freeFields** : configuration des informations complémentaires dans le tableau. Une information complémentaire se définit avec des balises obligatoires et des balises optionnelles.

! L'administrateur peut définir jusqu'à 3 informations complémentaires dans le tableau. Mettre un tableau vide « "freeFields": [] » pour aucune information complémentaire.

#### 6.4.2.2.1 Balises obligatoires d'une information complémentaire

Une information complémentaire se déclare avec les balises obligatoires suivantes :

- **name** : nom de l'attribut dans la base de données ;
- **description** : libellé de l'attribut dans le tableau ;
- **nullIndicator** : valeur à afficher dans le tableau si la valeur est nulle.

```
{  
  "name": "matériau",  
  "description": "Matériau",  
  "nullIndicator": "-"  
}
```

Figure 102 : Balises obligatoires d'une information complémentaire

#### 6.4.2.2.2 Balises facultatives d'une information complémentaire

Précision au niveau d'un formatage numérique :

- **numberFormat** : appliquer un formatage numérique (facultatif).
  - **digitsAfterZero** : nombre de chiffres après la virgule ;
  - **roundPolicy** : méthode d'arrondi
    - **round** : entier le plus proche (ex : 3,4 devient 3 et 3,6 devient 4) ;
    - **floor** : entier inférieur (ex : 3,4 devient 3 et 3,6 devient 3) ;
    - **ceil** : entier supérieur (ex : 3,4 devient 4 et 3,6 devient 4).

```
{  
  "name": "pente",  
  "description": "Pente en mm/m",  
  "nullIndicator": "-",  
  "numberFormat": {  
    "digitsAfterZero": 2,  
    "roundPolicy": "round"  
  }  
},
```

Figure 103 : Formatage numérique d'une information complémentaire

! Les balises « digitsAfterZero » et « roundPolicy » sont obligatoires si un formatage numérique est demandé.

Précision au niveau d'un formatage d'une date :

Les dates dans la base de données sont sauvegardées au format numériques, elles doivent être formatées pour l'affichage.

- **dateFormat** : appliquer un formatage date avec l'une des options suivantes (facultatif) :
  - **dd/MM/yyyy** (ex : 01/01/2025) ;
  - **dd-MM-yyyy** (ex : 01-01-2025) ;

- **dd/MM/yyyy HH:mm:ss** (ex : 01/01/2025 00:00:00) ;
- **HH:mm** (ex : 00:00).

```
{  
    "name": "datepose",  
    "description": "Date de pose",  
    "nullIndicator": "-",  
    "dateFormat": "dd-MM-yyyy"  
}
```

*Figure 104 : Formatage date d'une information complémentaire*

Concaténation de plusieurs champs :

Pour concaténer plusieurs champs la balise obligatoire « **name** » doit être renseignée avec la valeur « **systemConcat** ».

► **concatFormat** : appliquer une concaténation de plusieurs champs (facultatif).

- **names** : liste des attributs à concaténer ;
- **separator** : chaîne de caractères définissant le séparateur de champs (ex : "/").

```
{  
    "name": "systemConcat",  
    "description": "Hauteur/Largeur (mm)",  
    "nullIndicator": "-",  
    "concatFormat": {  
        "names": [  
            "hauteur",  
            "largeur"  
        ],  
        "separator": "/"  
    }  
},
```

*Figure 105 : Concaténation de champs dans une information complémentaire*

### 6.4.2.3 Tracé du profil

Le tracé du profil est positionné dans la partie supérieure du résultat :

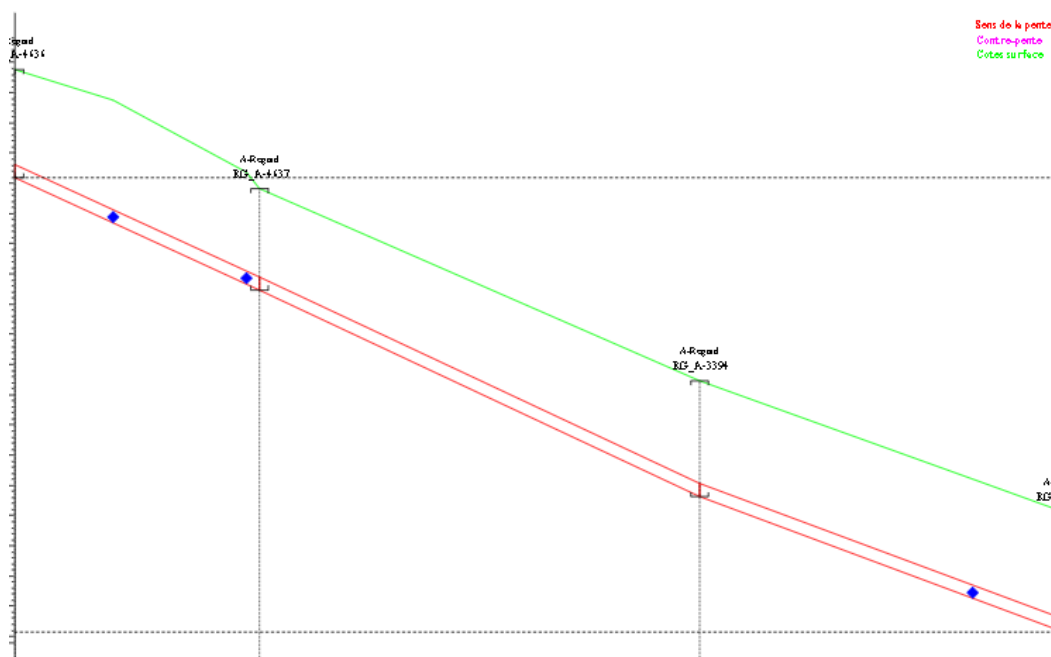


Figure 106 : Tracé du profil en long

#### 6.4.2.3.1 Définition des couleurs

La couleur d'un élément peut être définie de 2 façons :

- ▶ avec un code [couleur JAVA par son nom](#) (ex : red) ;
- ▶ avec un [code couleur HTML en hexadécimal](#) (ex : #ff0000).

Le paramétrage de la couleur s'applique au niveau des balises suivantes :

- ▶ **lineColor** : couleur d'une ligne ou d'un contour de polygone ;
- ▶ **fillColor** : couleur de remplissage d'un polygone ;
- ▶ **textColor** : couleur d'un texte.

```
"lineColor": "#D5DBDB",
"fillColor": "orange",
"textColor": "black",
```

Figure 107 : Définition des couleurs du profil en long

#### 6.4.2.3.2 Expressions de calcul des champs

Pour qu'une catégorie d'éléments soit tracée au niveau du profil, des informations au niveau de la base de données sont nécessaires (ex : tampon, radier, altitude, hauteur, largeur, longueur, ...).

Ces informations peuvent être récupérées :

- ▶ dans un attribut de l'élément (1) ;
- ▶ dans plusieurs champs, pouvant appartenir à un ou plusieurs éléments, en récupérant le premier renseigné (2) ;
- ▶ dans plusieurs champs, pouvant appartenir à un ou plusieurs éléments, en associant les différentes valeurs récupérées (3) ;
- ▶ à partir d'une combinaison de (2) et (3).

Le lien entre la catégorie d'éléments à tracer et la base de données est réalisé au niveau de la balise « **fieldExpressionsByPriority** ».

La balise « **fieldExpressionsByPriority** » contient une ou plusieurs expressions.

- **fieldExpressionsByPriority** : liste des expressions de calcul d'une information pour le tracé d'un élément du profil. Cette balise peut contenir une ou plusieurs expressions.
  - **operationType** : opérateur d'enchaînement des champs de l'expression. Les opérateurs possibles sont :
    - **OR** : prise en compte du premier champ renseigné ;
    - **ADD** : ajout des valeurs de tous les champs.
  - **fields** : liste des champs de l'expression définis par :
    - **source** : type d'élément avec les valeurs possibles suivantes :
      - **node** : pour un élément de type nœud (ponctuel) ;
      - **edge** : pour une élément de type arc (linéaire).
    - **name** : nom d'un champ de la source.
  - **unity** : utiliser cette balise, facultative, si la valeur dans la base de données n'est pas exprimée dans la mesure mentionnée au niveau de la balise « **defaultUnity** » (voir chapitre « [Paramètres généraux](#) »). Valeurs possibles :
    - **m** : mètre ;
    - **cm** : centimètre.
    - **mm** : millimètre.

```
"fieldExpressionsByPriority": [  
  {  
    "operationType": "ADD",  
    "fields": [  
      {  
        "source": "node",  
        "name": "radier"  
      },  
      {  
        "source": "node",  
        "name": "profond",  
        "unity": "mm"  
      }  
    ]  
  },  
  {  
    "operationType": "OR",  
    "fields": [  
      {  
        "source": "node",  
        "name": "cotetn"  
      },  
      {  
        "source": "node",  
        "name": "cotecalc"  
      }  
    ]  
  }  
]
```

Figure 108 : Expression d'un calcul au niveau du tracé

#### 6.4.2.3.3 Définition des libellés

Les libellés permettent d'afficher sur le profil, ou au survole d'un élément pour une info-bulle, son Asset Group (ex : A-Regard, A-Raccord de branchement) et son identifiant (ex : RG\_A-4637, RACB\_A-632).

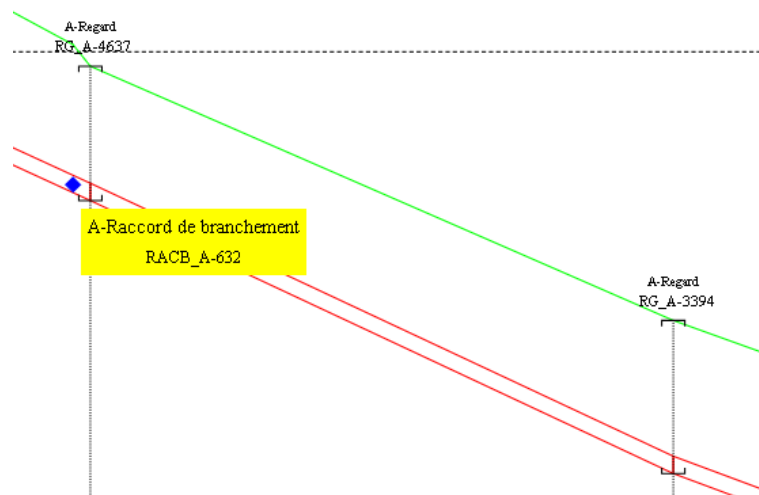


Figure 109 : Libellés sur un profil en long

L'affichage des libellés est géré par la balise « **label** » avec les options suivantes :

- **textColor** : permet de définir la [couleur](#) du texte du libellé.
- **showAssetGroupName** :
  - **true** : le libellé contient l'Asset Group ;
  - **false** : le libellé ne contient pas l'Asset Group.
- **labelIdentificationFields** : permet de définir comment l'identification de l'élément est récupéré par le biais d'une balise « **fieldExpressionsByPriority** ».

! Dans la version courante, le contenu de cette balise « **fieldExpressionsByPriority** » est imposé et doit être renseigné **obligatoirement** comme dans l'exemple ci-dessous.

```
"label": {
  "textColor": "black",
  "showAssetGroupName": true,
  "labelIdentificationFields": {
    "fieldExpressionsByPriority": [
      {
        "operationType": "OR",
        "fields": [
          {
            "source": "node",
            "name": "ident"
          }
        ]
      }
    ]
  }
},
```

Figure 110 : Définition d'un libellé du profil en long

#### 6.4.2.3.4 Définition des symboles

Certains éléments sont représentés par des symboles. La gestion de leur représentation est réalisée au niveau de la balise « **symbol** » avec les options suivantes :

- ▶ **type** : seul le symbole « **diamond** » (losange) est proposé actuellement.
- ▶ **lineColor** : [couleur](#) de contour du symbole
- ▶ **fillColor** : [couleur](#) de remplissage du symbole.
- ▶ **opacity** : transparence du symbole. Valeur décimale entre 0 (transparent) et 1 (opaque).

```
"symbol": {  
  "type": "diamond",  
  "lineColor": "cyan",  
  "fillColor": "cyan",  
  "opacity": 0.4  
}
```

Figure 111 : Définition d'un symbole du profil en long

#### 6.4.2.3.5 Tracé des linéaires

La définition du tracé des linéaires est gérée au niveau de la balise « **drawPipeSection** ».

```
"drawPipeSection": {  
  "slope": {  
    "counterSlope": {  
      "pipeZFields": {  
        "pipeDiameterFields": {  
          "pipeLengthFields": {  
            }  
          }  
        }  
      }  
    }  
  },  
}
```

Figure 112 : Paramétrage du tracé des linéaires pour le profil

Les options de tracé sont les suivantes :

- ▶ **slope** : option de tracé d'un linéaire dans le sens de la pente pour un réseau gravitaire (assainissement) ou d'un linéaire dans n'importe quel sens pour un réseau sous pression (eau).
  - **lineColor** : voir « [Définition des couleurs](#) ».

```
"slope": {  
  "lineColor": "red"  
},
```

Figure 113 : Couleur du tracé de la pente du profil

- ▶ **counterSlope** : option de tracé d'un linéaire dans le sens opposé à la pente (facultatif). Cette balise doit être utilisée seulement pour un réseau gravitaire.
  - **lineColor** : voir « [Définition des couleurs](#) ».

```
"counterSlope": {  
  "lineColor": "magenta"  
},
```

Figure 114 : Couleur de tracé de la contre-pente du profil

- ▶ **pipeZFields** : principe de récupération de la cote fil d'eau des linéaires au niveau de chaque nœud.
  - **fromExtremity** : au niveau du nœud associé au premier point du linéaire ;
  - **toExtremity** : au niveau du nœud associé au dernier point du linéaire.

```
"pipeZFields": {
  "fromExtremity": {
    "fieldExpressionsByPriority": [
      {
        "operationType": "OR",
        "fields": [
          {
            "source": "edge",
            "name": "fromz"
          },
          {
            "source": "node",
            "name": "radier"
          }
        ]
      }
    ]
  },
  "toExtremity": {
    "fieldExpressionsByPriority": [
      {
        "operationType": "OR",
        "fields": [
          {
            "source": "edge",
            "name": "toz"
          },
          {
            "source": "node",
            "name": "radier"
          }
        ]
      }
    ]
  }
},
```

*Figure 115 : Paramétrage pour récupérer la cote fil d'eau des linéaires*

La définition des balises « **fromExtremity** » et « **toExtremity** » indique le principe pour déterminer la cote d'eau des linéaires. Dans l'exemple ci-dessus, l'information est récupérée en priorité au niveau du linéaire (« **edge** ») sur l'attribut associé à la balise « **name** » (ex : fromz ou toz). Si la valeur n'est pas renseignée au niveau du linéaire, elle est récupérée au niveau du nœud (« **node** »), en coïncidence géographique, dans l'attribut qui est indiqué au niveau de la balise « **name** » (ex : radier).

La balise « **pipeZFields** » permet de tracer le linéaire qui est représenté en rouge dans l'exemple ci-dessous.



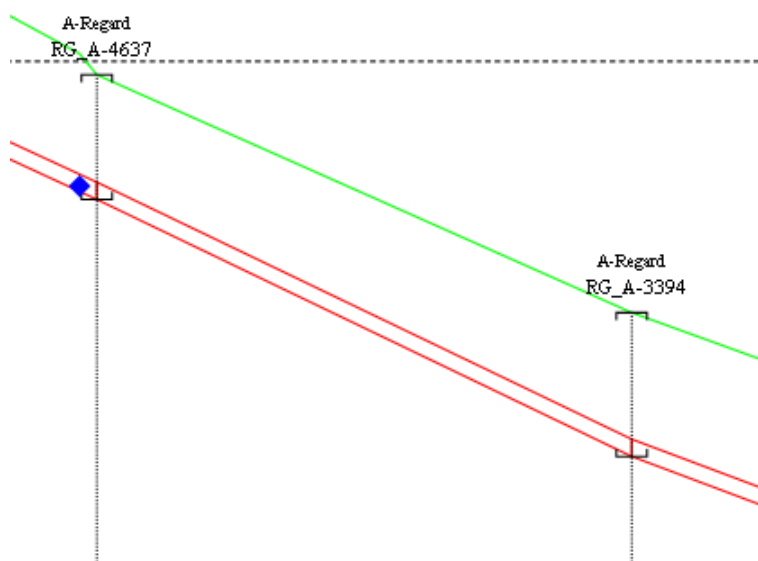


Figure 116 : Tracé d'un linéaire utilisant la cote radier des regards

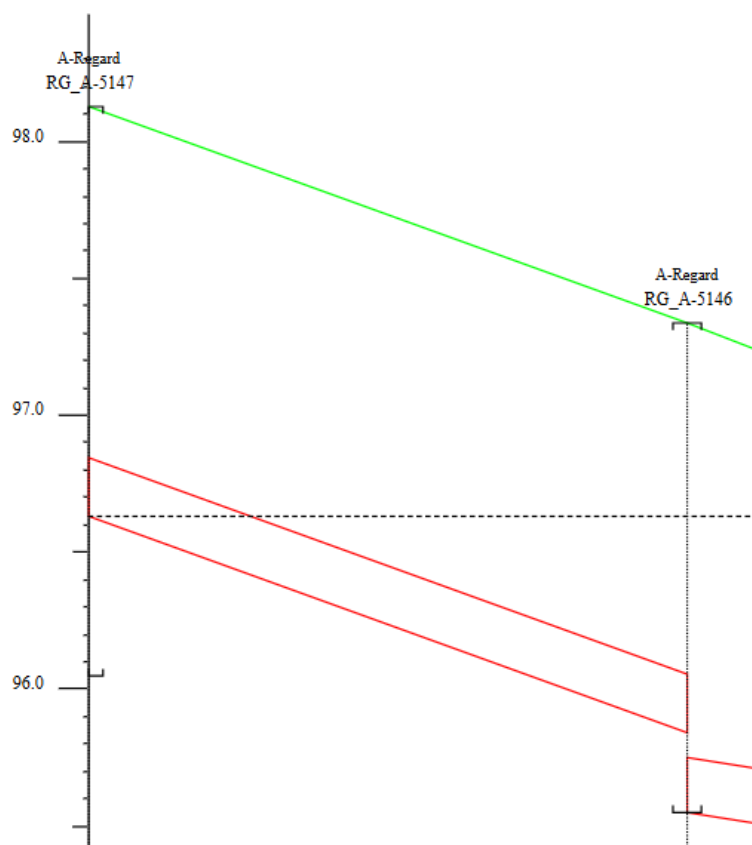


Figure 117 : Tracé d'un linéaire utilisant ses côtés fil d'eau

► **pipeDiameterFields** : principe de récupération de la dimension (hauteur ou diamètre) du linéaire.

! La balise « **source** » doit toujours être renseignée avec la valeur « **edge** ».

```

"pipeDiameterFields": {
  "fieldExpressionsByPriority": [
    {
      "operationType": "OR",
      "fields": [
        {
          "source": "edge",
          "name": "diametre",
          "unity": "mm"
        },
        {
          "source": "edge",
          "name": "diamext",
          "unity": "mm"
        }
      ]
    }
  ]
},

```

**Figure 118 : Paramétrage pour récupérer la dimension des linéaires**

Dans l'exemple ci-dessus, la dimension du linéaire est lue en priorité au niveau de l'attribut « diametre » puis, si celui-ci n'est pas renseigné, au niveau de l'attribut « diamext ». Si aucun de ces 2 attributs n'est renseigné, le linéaire sera représenté par une simple ligne entre des côtes fil d'eau.

► **pipeLengthFields** : principe de récupération de la longueur du linéaire.

! La balise « **source** » doit toujours être renseignée avec la valeur « **edge** ».

```

"pipeLengthFields": {
  "fieldExpressionsByPriority": [
    {
      "operationType": "OR",
      "fields": [
        {
          "source": "edge",
          "name": "longueur"
        }
      ]
    }
  ]
},
!

```

**Figure 119 : Paramétrage pour récupérer la longueur des linéaires**

Dans l'exemple ci-dessus, la longueur du linéaire est lue en priorité au niveau de l'attribut « longueur » et, si celui-ci n'est pas renseigné, la longueur géométrique sera utilisée.

! Comme pour la dimension (pipeDiameterFields), il est possible de définir plusieurs champs avec une condition « OR » avant de récupérer la longueur géométrique.

#### 6.4.2.3.6 Tracé du terrain

Le tracé de la ligne de surface est géré par la balise « **drawTerrainSection** ». Le tracé de la ligne de surface est calculé en parcourant chaque linéaire du profil pour conserver le même ordre et le même sens de tracé entre les 2 lignes (linéaire et terrain).

```
"drawTerrainSection": {  
  "lineColor": "green",  
  "terrainZFields": {  
    "primaryFromNode": {  
      "auxiliaryFromEdge": {  
      }  
    }  
  },  
}
```

Figure 120 : Paramétrage du tracé du terrain

Les options de tracé sont les suivantes :

- **lineColor** : permet de définir la [couleur](#) des traits du terrain.
- **terrainZFields** : permet de définir le principe de récupération des différents points d'altitude pour tracer la ligne de surface.

La balise « **terrainZFields** » permet de tracer la ligne de surface en vert dans l'exemple ci-dessous.

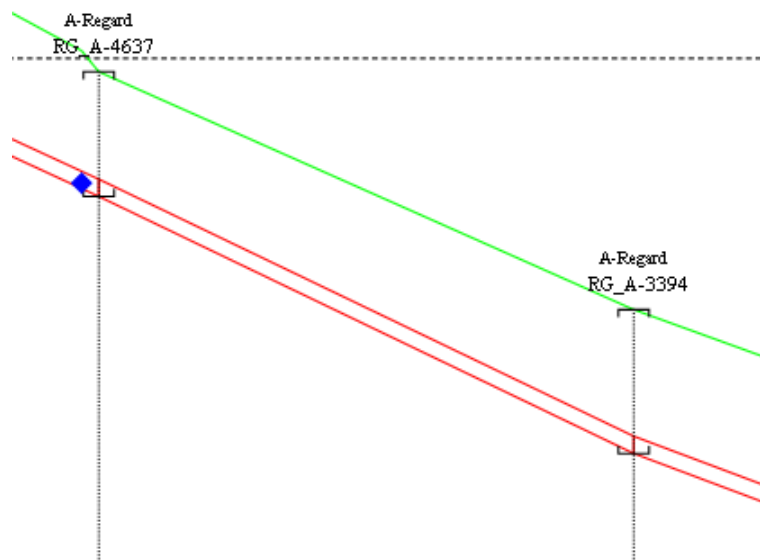


Figure 121 : Exemple de tracé du terrain basé sur le tampon des regards

La balise « **terrainZFields** » contient les options suivantes :

- **primaryFromNode** : permet de définir le(s) champ(s) des ponctuels (node) à utiliser pour déterminer l'altitude du terrain.

! La balise « **source** » doit toujours être renseignée avec la valeur « **node** ».

```

"primaryFromNode": {
  "fieldExpressionsByPriority": [
    {
      "operationType": "ADD",
      "fields": [
        {
          "source": "node",
          "name": "radier"
        },
        {
          "source": "node",
          "name": "profond"
        }
      ]
    },
    {
      "operationType": "OR",
      "fields": [
        {
          "source": "node",
          "name": "cotetn"
        }
      ]
    }
  ]
},
! },

```

**Figure 122 : Paramétrage pour récupérer la côte de surface à partir des ponctuels**

Dans l'exemple ci-dessus, l'altitude au niveau du ponctuel (node) est déterminée, en priorité, par la somme des champs « **radier** » et « **profond** » en coïncidence géographique avec le linéaire courant. Si au moins l'un des 2 champs n'est pas renseigné, l'altitude est récupérée au niveau du champ « **cotetn** ». Si la valeur de l'altitude n'a toujours été récupérée le traitement bascule au niveau de la balise « **auxiliaryFromEdge** » si elle est définie.

- **auxiliaryFromEdge** : permet de définir le(s) champ(s) des linéaires (edge) à utiliser pour déterminer l'altitude du terrain (facultatif).
  - **fromExtremity** : au niveau du premier point du linéaire ;
  - **toExtremity** : au niveau du dernier point du linéaire.
- ! La balise « **source** » doit toujours être renseignée avec la valeur « **edge** ».
- ! Cette balise est exploitée seulement si l'altitude n'a pas pu être récupérée au niveau des ponctuels (primaryFromNode).

```

    "auxiliaryFromEdge": {
      "fromExtremity": {
        "fieldExpressionsByPriority": [
          {
            "operationType": "ADD",
            "fields": [
              {
                "source": "edge",
                "name": "fromz"
              },
              {
                "source": "edge",
                "name": "fromprf"
              }
            ]
          }
        ]
      },
      "toExtremity": {
    }
  }

```

**Figure 123 : Paramétrage pour récupérer la côté de surface au niveau du premier point**

```

    "auxiliaryFromEdge": {
      "fromExtremity": {
        "toExtremity": {
          "fieldExpressionsByPriority": [
            {
              "operationType": "ADD",
              "fields": [
                {
                  "source": "edge",
                  "name": "toz"
                },
                {
                  "source": "edge",
                  "name": "toprf"
                }
              ]
            }
          ]
        }
      }
    }
  }

```

**Figure 124 : Paramétrage pour récupérer la côté de surface au niveau du dernier point**

La définition des balises « **fromExtremity** » et « **toExtremity** » indique le principe pour déterminer la côté terrain à partir des linéaires.

Dans les exemples ci-dessus, l'information est récupérée par la somme de la côté fil d'eau (« **fromz** » ou « **toz** ») et de la profondeur de la côté fil d'eau (« **fromprf** » ou « **toprf** »).

! Si les informations pour tracer la ligne de surface n'ont pas pu être récupérées, la ligne de surface n'est pas tracée au niveau du résultat.

#### 6.4.2.3.7 Tracé des ponctuels en extrémité

Les ponctuels en extrémité sont des nœuds coupants qui séparent plusieurs linéaires. Le raccord de jonction et la vanne, dans l'exemple ci-dessous, sont des ponctuels en extrémité.

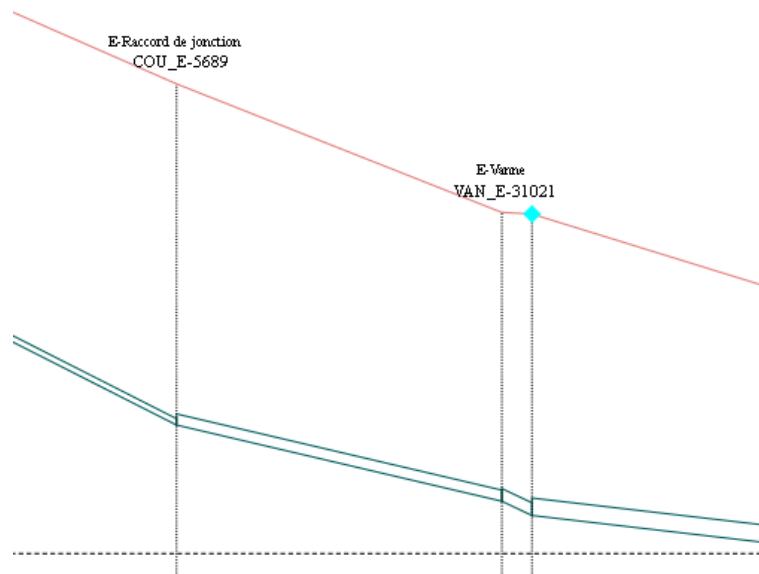


Figure 125 : Exemple tracé des ponctuels en extrémité

Le tracé des ponctuels d'extrémité est géré par la balise « **drawExtremityNodeSection** ». Cette balise contient 2 sections communes et 1 section spécifique en fonction du métier (eau potable ou assainissement).

#### 6.4.2.3.7.1 Sections communes

```
"drawExtremityNodeSection": {  
  "hasVerticalDottedLine": true,  
  "label": {  
    "labelOverlapAlternative": {  
      "symbol": {  
      },  
    },  
  },  
}
```

Figure 126 : Paramétrage du tracé des ponctuels en extrémité

- **hasVerticalDottedLine** : indique si la ligne verticale en pointillées, au niveau de chaque ponctuel d'extrémité, doit être tracée. Les valeurs possibles sont « **true** » ou « **false** ».
- **label** : permet de définir le [libellé du ponctuel d'extrémité](#) à afficher.
- **labelOverlapAlternative** : Si plusieurs libellés se recouvrent, du fait de la proximité des ponctuels, seul le libellé du premier ponctuel de la « zone » est dessiné. Pour tous les autres, un symbole est affiché et il est [défini](#) par la section « **labelOverlapAlternative** ».

#### 6.4.2.3.7.2 Métier Eau

La balise « **extremityNodeZFields** » permet de définir le tracé des ponctuels d'extrémité, au niveau de la ligne de surface, pour le métier de l'eau.

```

"extremityNodeZFields": {
  "surface": {
    "fieldExpressionsByPriority": [
      {
        "operationType": "OR",
        "fields": [
          {
            "source": "node",
            "name": "cotetn"
          }
        ]
      }
    ]
  }
}

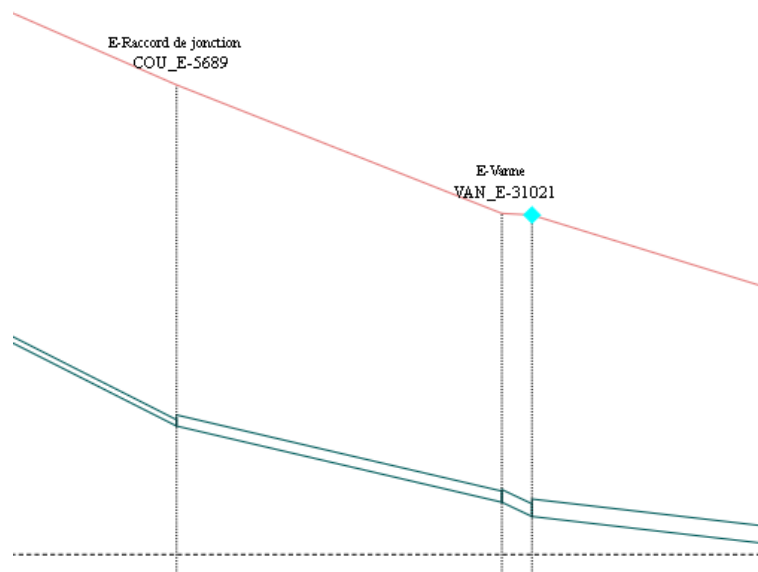
```

**Figure 127 : Paramétrage du tracé des ponctuels d'extrémité (Eau)**

La balise « **extremityNodeZFields** » contient une balise « **surface** » puis une liste associée à la balise « **fieldExpressionsByPriority** » contenant les champs à exploiter pour le tracé.

Dans l'exemple de paramétrage ci-dessus, la valeur du champ « cotetn » est utilisé, si elle est définie, pour déterminer la position du libellé et, éventuellement, tracer le trait de rappel vertical associé au ponctuel d'extrémité.

! La balise « **source** » doit toujours être renseignée avec la valeur « **node** ».



**Figure 128 : Exemple d'un tracé de ponctuels d'extrémité (Eau)**

#### 6.4.2.3.7.3 Métier Assainissement

```
"manhole": {  
  "low": {  
    "symbolColor": "black",  
    "manholeZFields": {  
    },  
  },  
  "high": {  
    "symbolColor": "black",  
    "manholeZFields": {  
    }  
  }  
}
```

Figure 129 : Paramétrage du tracé des ponctuels d'extrémité (Assainissement)

La balise « **manhole** » permet de définir le tracé des regard en extrémité pour le métier de l'assainissement. Cette balise permet ensuite de définir 2 parties :

- **low** : permet de définir le tracé au niveau du radier ;
- **high** : permet de définir le trace au niveau du tampon.

Chacune de ces 2 parties est paramétrable avec :

- **symbolColor** : permet de définir la [couleur du symbole](#) ;
- **manholeZFields** : permet de définir la méthode pour récupérer l'information d'altitude.

! Dans la partie « **manholeZFields** », la balise « **source** » doit toujours être renseignée avec la valeur « **node** ».

```
"low": {  
  "symbolColor": "black",  
  "manholeZFields": {  
    "fieldExpressionsByPriority": [  
      {  
        "operationType": "OR",  
        "fields": [  
          {  
            "source": "node",  
            "name": "radier"  
          }  
        ]  
      }  
    ]  
  }  
},
```

Figure 130 : Paramétrage de la partie radier du regard (Assainissement)

L'exemple ci-dessus, avec la balise « **low** », récupère la valeur du champ « **radier** » pour tracer le symbole en couleur noire au niveau du radier du regard. Si la valeur n'est pas renseignée, il n'est pas dessiné.



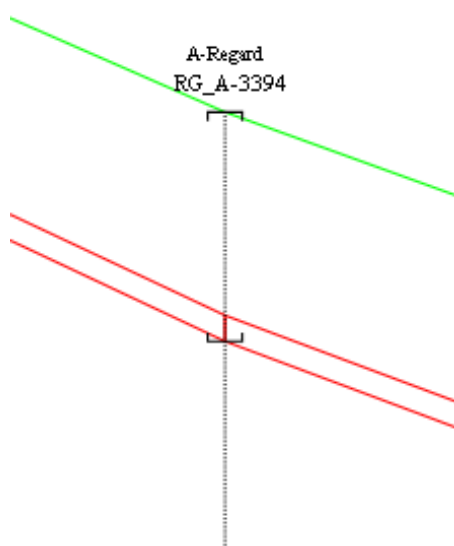
```

"high": {
  "symbolColor": "black",
  "manholeZFields": {
    "fieldExpressionsByPriority": [
      {
        "operationType": "ADD",
        "fields": [
          {
            "source": "node",
            "name": "radier"
          },
          {
            "source": "node",
            "name": "profond"
          }
        ]
      },
      {
        "operationType": "OR",
        "fields": [
          {
            "source": "node",
            "name": "cotetn"
          }
        ]
      }
    ]
  }
}

```

**Figure 131 : Paramétrage de la partie tampon du regard (Assainissement)**

L'exemple ci-dessus, avec la balise « **high** », décrit le principe pour tracer le symbole en couleur noir, le libellé ainsi que le trait de rappel vertical au niveau du tampon du regard. L'altitude du tampon est récupérée en sommant les champs « **radier** » et « **profond** ». Si au moins l'un de ces champs n'est pas renseigné, la valeur est récupérée dans le champ « **cotetn** ». Si la valeur n'est toujours pas renseignée, les éléments de surface du tampon (symbole, libellé et trait) ne sont pas représentés.



**Figure 132 : Exemple de tracé d'un regard (Assainissement)**

#### 6.4.2.3.8 Tracé des ponctuels en vertex

Les ponctuels en vertex sont des éléments non coupants, connecté sur un linéaire, qui ne nécessite pas, dans le version actuelle, de données pour définir leur altitude.

Le représentation des ponctuels en vertex est géré par la balise « **drawVertexNodeSection** ».

```
"drawVertexNodeSection": {  
  "symbol": {  
    "label": {  
      }  
    }  
  }  
}
```

Figure 133 : Paramétrage du tracé des ponctuels en vertex

La balise « **drawVertexNodeSection** » contient les options suivantes :

- **symbol** : permet de définir le [symbole](#) ;
- **label** : permet de définir l'[info-bulle](#) du ponctuel en vertex lors de son survol par le curseur.

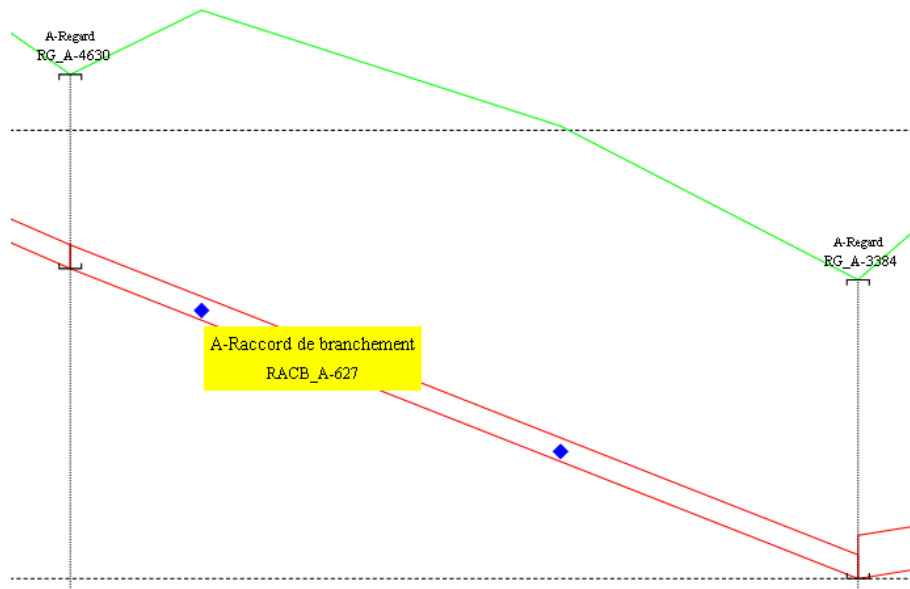


Figure 134 : Exemple d'info-bulle au niveau d'un raccord de branchement

#### 6.4.2.3.9 Modèle numérique de terrain (MNT)

La ligne de surface peut être représentée soit en utilisant les informations attributaires d'altitude soit en exploitation un modèle numérique de terrain si celui-ci a été configuré.

La configuration du MNT se fait dans le groupe « **mnt** » :

```
"mnt": {  
  "densifyStep": 1,  
  "active": false  
}
```

Figure 135 : Paramétrage MNT du module « Profil en long »

- **densifyStep** : distance en mètres terrain entre 2 mesures de hauteur;
- **active** : true pour activer l'option MNT ou false.

- 
- ! Pour que l'option MNT, au niveau du module « Profil en long », soit disponible, il faut également configurer la source au niveau de la balise « **mnt** » dans le fichier « **config.json** » du ou des métiers concernés.