



## 1Water

---

# Guide Administrateur

## Version 3.2

16/12/2024

## 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 © 2024, 1Spatial. Tous droits réservés.

## TABLE DES MATIERES

<b>1</b>	<b>INTRODUCTION .....</b>	<b>8</b>
<b>2</b>	<b>CONFIGURATION GENERALE.....</b>	<b>9</b>
2.1	Licences.....	9
2.2	Portails .....	9
2.3	Serveurs .....	9
2.4	Utility Network.....	10
<b>3</b>	<b>CONFIGURATION DES APPLICATIONS .....</b>	<b>11</b>
3.1	Eau potable .....	11
3.1.1	General.....	11
3.1.2	Anomalie .....	19
3.1.3	Coupe d'eau.....	21
3.1.4	Export de données .....	25
3.1.5	Recherche de fuite .....	26
3.1.6	Notation .....	26
3.1.7	Mise à jour.....	28
3.1.8	Cadastre .....	31
3.1.9	Gestion des conséquences .....	33
3.2	Assainissement .....	34
3.2.1	General.....	34
3.2.2	Anomalie .....	43
3.2.3	Encrassement .....	45
3.2.4	Point noir.....	48
3.2.5	Export de données .....	50
3.2.6	Propagation .....	50
3.2.7	Profil en long .....	51
3.2.8	Inspection télévisée .....	53
3.2.9	Intervention.....	61
3.2.10	Notation .....	65
3.2.11	Mise à jour.....	67
3.2.12	Cadastre .....	70
3.2.13	Gestion des conséquences .....	72
<b>4</b>	<b>CONFIGURATION DE LA MOBILITE.....</b>	<b>74</b>
4.1	Fichiers de configuration .....	74
4.2	Gestion des filtres .....	74
4.2.1	Condition Simple .....	74
4.2.2	Filtre simple .....	77
4.2.3	Filtre complexe.....	78
4.3	Configuration de la recherche par adresse.....	80
4.4	Configuration du module « Graffiti ».....	81

---

<b>5 UTILITAIRES.....</b>	<b>82</b>
5.1 Export EPANET .....	82
5.1.1 Fichier de paramétrage .....	82
5.1.2 Exemple .....	90
5.2 Export SWMM.....	90
5.2.1 Fichier de paramétrage .....	91
5.2.2 Exemple .....	98
5.3 Export StaR-DT.....	99
5.3.1 Définition.....	100
5.3.2 Exporter les données du réseau au format StaR-DT .....	100
5.3.3 Modèle conceptuel de données.....	100
5.3.4 Correspondance 1Water .....	101
5.3.5 Fichier de paramétrage .....	101
5.4 Méthode de notation.....	107
5.4.1 Note.....	108
5.4.2 Critère .....	109
<b>6 ANNEXES .....</b>	<b>123</b>
6.1 Filtres de sélection personnalisés .....	123
6.1.1 Généralités .....	123
6.1.2 Assainissement.....	123
6.1.3 Eau.....	125
6.2 Définir les cartes .....	126

## FIGURES

Figure 1 : Informations sur le réseau de distribution (Eau).....	12
Figure 2 : Localisation (Eau).....	12
Figure 3 : Localisation – Restriction de la zone de recherche (Eau).....	12
Figure 4 : Impression (Eau).....	14
Figure 5 : Couches d'exploitation (Eau).....	15
Figure 6 : Couches de sectorisation (Eau) .....	16
Figure 7 : Couches de dépose (Eau) .....	16
Figure 8 : Rapports (Eau) .....	17
Figure 9: Service d'impression (Eau) .....	17
Figure 10 : Anomalie (Eau) .....	19
Figure 11 : Patrimoine pour anomalie (Eau) .....	20
Figure 12 : Coupure - Trace .....	22
Figure 13 : Coupure - Filtres .....	22
Figure 14 : Coupure - Éléments impactés .....	24
Figure 15 : Export (Eau) .....	25
Figure 16 : Recherche de fuites.....	26
Figure 17 : Notation (Eau) .....	26
Figure 18 : Définition des notes (Eau) .....	28
Figure 19 : Prototype « Point de livraison » .....	29
Figure 20 : Configuration du cadastre sous PostgreSQL (Eau) .....	31
Figure 21 : Informations sur le réseau de distribution (Assainissement) .....	34
Figure 22 : Localisation (Assainissement).....	35
Figure 23 : Localisation - Restriction de la zone de recherche (Assainissement).....	35
Figure 24 : Impression (Assainissement).....	36
Figure 25 : Couches d'exploitation (Assainissement).....	37
Figure 26 : Couches de sectorisation (Assainissement) .....	39
Figure 27 : Rapports (Assainissement) .....	40
Figure 28 : Service d'impression (Assainissement) .....	41
Figure 29 : Anomalie (Assainissement) .....	43
Figure 30 : Patrimoine pour anomalie (Assainissement) .....	44
Figure 31 : Encrassement .....	46
Figure 32 : Point noir .....	48
Figure 33 : Export (Assainissement) .....	50
Figure 34 : Propagation (Assainissement 2 domaines) .....	51
Figure 35 : Propagation (Assainissement 1 domaine).....	51
Figure 36 : Trace du profil en long (Assainissement) .....	52
Figure 37 : Rapport du profil en long (Assainissement) .....	52
Figure 38 : MNT pour profil en long (Assainissement) .....	53
Figure 39 : Généralités ITV .....	53
Figure 40 : Patrimoine pour les inspections du réseau .....	55

Figure 41 : Filtres attributaires pour les inspections du réseau .....	56
Figure 42 : Trace pour les inspections du réseau .....	57
Figure 43 : Généralités pour intervention (Assainissement) .....	61
Figure 44 : Patrimoine pour les interventions (Assainissement) .....	62
Figure 45 : Filtres attributaires pour les interventions (Assainissement) .....	63
Figure 46 : Trace pour les interventions (Assainissement) .....	65
Figure 47 : Patrimoine pour la notation (Assainissement) .....	65
Figure 48 : Définition des notes (Assainissement) .....	67
Figure 49 : Prototype « Point de collecte (eaux usées) » .....	68
Figure 50 : Configuration du cadastre sous PostgreSQL (Assainissement) .....	70
Figure 51 : Partie « headers » .....	83
Figure 52 : Partie « options » .....	83
Figure 53 : Partie "domains" .....	83
Figure 54 : Entité d'une section .....	84
Figure 55 : Section "SYSTEM" .....	84
Figure 56 : Filtre sur une entité .....	84
Figure 57 : Valeur avec dictionnaire et « next » .....	85
Figure 58 : Valeur opération de comparaison et plusieurs « next » .....	85
Figure 59 : Exemple de dictionnaire avec un opérateur de comparaison d'égalité implicite .....	86
Figure 60 : Exemple de dictionnaire avec un opérateur de comparaison strictement inférieur .....	86
Figure 61 : Exemple d'un modèle d'attributs .....	87
Figure 62 : Exemple d'un modèle conditionnel d'attributs .....	88
Figure 63 : Partie « Paramètres » .....	89
Figure 64 : Exemple de la section « PIPES » .....	90
Figure 65 : Partie « headers » .....	91
Figure 66 : Partie "options" .....	92
Figure 67 : Partie "domains" .....	92
Figure 68 : Entité d'une section .....	93
Figure 69 : Section "SYSTEM" .....	93
Figure 70 : Filtre sur une entité .....	93
Figure 71 : Valeur avec dictionnaire et « next » .....	94
Figure 72 : Valeur opération de comparaison et plusieurs « next » .....	94
Figure 73 : Exemple de dictionnaire avec un opérateur de comparaison d'égalité implicite .....	95
Figure 74 : Exemple de dictionnaire avec un opérateur de comparaison strictement inférieur .....	95
Figure 75 : Exemple d'un modèle d'attributs .....	96
Figure 76 : Exemple d'un modèle conditionnel d'attributs .....	97
Figure 77 : Partie « Paramètres » .....	98
Figure 78 : Exemple de la section « OUTFALLS » .....	98
Figure 79 : Exemple de modèle d'attributs « OUTFALLS » .....	99
Figure 80 : Partie « headers » .....	102
Figure 81 : Contenu de « headers » .....	102
Figure 82 : Partie « Paramètres » .....	102
Figure 83 : Exemple de dictionnaire avec un opérateur de comparaison d'égalité implicite .....	103

---

Figure 84 : Exemple de dictionnaire avec un opérateur de comparaison supérieur ou égal .....	103
Figure 85 : Partie "domains" .....	104
Figure 86 : Entité d'une section.....	104
Figure 87 : Filtre sur une entité .....	104
Figure 88 : Valeur avec dictionnaire et « next ».....	105
Figure 89 : Exemple d'un modèle d'attributs .....	106
Figure 90 : Concaténation de 2 valeurs.....	107
Figure 91 : Rechercher une valeur dans un dictionnaire .....	107

## 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 GENERALE

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** ».

## 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": "de50121f503143d69e2a10206ad779f4",
    "networkService": "default:/1WATER_ORCL/1SOrc1Water",
    "exploitationService": "default:/1WATER_ORCL/1SOrc1Exploitation",
    "sectorizationService": "default:/1WATER_ORCL/1SOrc1Sectorisation",
    "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 ;
- **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",
      "options": {
        "postcode": 56800
      },
      "featureServiceUrl": "default:/1WATER_ORCL/1SOrc1Water",
      "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": 56800
      },
      "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 renvoyés 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,  
    "measureLayerId": 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 ;

- ▶ **measureLayerId** : 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 dépôse 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 dépôse (Eau)*

- ▶ **wRemovalZone**: identifiant de la couche de zonage des dépôses ;
- ▶ **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": {  
    "equipment": {  
      "filepath": "EQPT_EAU"  
    },  
    "cutoffNotificationPattern": {  
      "filepath": "RAPPORT_ARRET/NOTIFICATION_MAIL_STD_PATTERN"  
    },  
    "cutoffAnnouncement": {  
      "filepath": "RAPPORT_ARRET/AVIS_ARRET"  
    },  
    "cutoff": {  
      "filepath": "CUTOFF"  
    },  
    "deviceSynthesis": {  
      "filepath": "EQPT_EAU"  
    }  
  },
```

Figure 8 : Rapports (Eau)

- ▶ **equipment** : Génère une fiche de vie de l'équipement. Champ obsolète, gardé pour compatibilité ascendante. Il est remplacé par « **deviceSynthesis** » :
  - **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.
- ▶ **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.

### 3.1.1.6 Service d'impression

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

```
  "printService": {  
    "url": "default:/Utilities/PrintingTools/GPSServer/Export%20Web%20Map%20Task",  
    "backgroundLayer": {  
      "styleUrl": "https://basemaps.arcgis.com/arcgis/rest/services/World_Basemap_v2/VectorTileServer/resources/styles/root.json",  
      "id": "178d9874f85-layer-1"  
    }  
  },
```

Figure 9: Service d'impression (Eau)

- ▶ **url** : Url vers le service d'impression ;
- ▶ **backgroundLayer**: Informations sur la couche de fond de plan à utiliser lors de l'impression. Seule une couche de tuiles vectorielles peut être renseignée (provenant de ArcGIS Pro) :
  - **styleUrl**: URL absolue vers le style du service de tuiles vectorielles à utiliser ;
  - **Id** : Identifiant de la couche à utiliser.

### 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 * * *"  
},
```

- **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 * * *"  
},
```

- **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.

### 3.1.2 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 géographie 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 10 : 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 11 : 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 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 :

- o **source** : type d'usage de l'équipement au niveau du réseau utility network.  
Valeurs possibles : « esriUNFCUTDevice » | « esriUNFCUTLine » | « esriUNFCUTJunction » ;
  - o **assetGroup** : code de l'assetGroup correspondant à l'équipement au sein de la couche concernée ;
  - o **assetType** : code de l'assetType correspondant au type d'équipement de l'assetGroup concerné. Exemple : tronçon (=assetGroup) de type refoulement, décharge, simple, etc... (=assetType) ;
  - o **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 12 : 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 13 : 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 linaire 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 14 : 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 15 : 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/GPSServer/ListenerCover/execute"
}
```

Figure 16 : 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 Notation

#### 3.1.6.1 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 17 : 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).

### 3.1.6.2 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 18 : Définition des notes (Eau)*

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

### 3.1.7 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": "PDL_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 19 : 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": [
            {
              "source": "esriUNFCUTDevice",
              "name": "Création d'un appareil de protection - Ventouse",
              "steps": [
                "features": [
                ]
              ]
            }
          ]
        ]
      }
    ]
  }
},
```

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

### 3.1.8 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 20 : 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: <xMIRx5J19lnuoTO+0Pr2zx8N2YE7L2QPz1T+k03+Vo=>
```

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

Exemple : `cadastre.database.password=~xMIRx5J19lnuoTO+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.

### 3.1.9 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_config.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 valeur de l'attribut est toujours remplacée (valeur par défaut) ;
      - 1 : la valeur de l'attribut est remplacée si elle n'est pas renseignée ;
      - 2 : si la valeur de l'attribut est nulle, elle est remplacée par la valeur constatée. Sinon, un traitement manuel est nécessaire.

## Exemple de fichier de configuration de gestion des conséquences :

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

```
"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": "a9c9e8fede864978bf5b166eaa24b876",
    "networkService": "default:/1WATER_ORCL/1SOrc1Sanitation",
    "exploitationService": "default:/1WATER_ORCL/1SOrc1Exploitation",
    "sectorizationService": "default:/1WATER_ORCL/1SOrc1Sectorisation",
    "domainNetworkNames": [
      "RES_EP",
      "RES_EU"
    ]
  }
]
```

*Figure 21 : Informations sur le réseau de distribution (Assainissement)*

- **name** : Nom du réseau ;
- **label** : libellé qui s'affiche dans l'IHM ;
- **webmap** : identifiant de la carte ;
- **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** » :

**Figure 23 : Localisation - Restriction de la zone de recherche (Assainissement)**

**Figure 22 : Localisation (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 renournés 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 24 : 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 25 : 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 26 : 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 dépôse 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 dépôses 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": {  
    "equipment": {  
        "filepath": "EQPT_ASS"  
    },  
    "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"  
    }  
},
```

Figure 27 : Rapports (Assainissement)

- **equipment** : Génère une fiche de vie de l'équipement. Champ obsolète, gardé pour compatibilité ascendante. Il est remplacé par « **deviceSynthesis** ».
- **filepath** : Nom du fichier qui permet de générer le rapport précédent du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.
- **tvi** : Génère une synthèse d'une inspection télévisée.
- **filepath** : Nom du fichier qui permet de générer le rapport précédent 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 précédent 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 précédent 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 précédent 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 précédent 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 précédent 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 précédent 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 précédent 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 précédent du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.
- **equipment** : Génère une fiche de vie de l'équipement.
  - **filepath** : Nom du fichier qui permet de générer le rapport précédent du chemin relatif s'il n'est pas placé à la racine du répertoire des rapports.

### 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",
  "backgroundLayer": {
    "styleUrl": "https://basemaps.arcgis.com/arcgis/rest/services/World_Basemap_v2/VectorTileServer/resources/styles/root.json",
    "id": "178d9874f85-layer-1"
  }
},
```

*Figure 28 : Service d'impression (Assainissement)*

- **url** : Url vers le service d'impression ;
- **backgroundLayer**: Informations sur la couche de fond de plan à utiliser lors de l'impression. Seule une couche de tuiles vectorielles peut être renseignée (provenant de ArcGIS Pro) :
  - **styleUrl**: URL absolue vers le style du service de tuiles vectorielles à utiliser ;
  - **Id** : Identifiant de la couche à utiliser.

### 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 liés 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 * * *"  
},
```

- **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 * * *"  
},
```

- **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.

### 3.2.2 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 géographie 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 29 : 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 30 : 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 géographie 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 31 : 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 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 géographie 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 32 : 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 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).
- **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": "SWMM",
      "url": "default://{{sanitation.exportServices.SWMUrl}}"
    },
    {
      "name": "STARDT",
      "url": "default://{{sanitation.exportServices.STARDTUrl}}"
    }
  ]
}
```

Figure 33 : 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 34 : Propagation (Assainissement 2 domaines)

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

Figure 35 : 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 ».

#### 3.2.7.1 Configuration de la trace

Un profil en long s'appuie sur une trace « chemin le plus court » entre deux points du réseau d'un « 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_SANITATION_LONG_PROFIL",
"startingPointCategoryNames" : ["LONG_PROFIL_LINEAR"]
```

Figure 36 : Trace du profil en long (Assainissement)

- ▶ **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.

### 3.2.7.2 Configuration du rapport

La configuration du rapport est définie dans le groupe « report » :

```
"report": {
  "dateFormat": "dd-MM-yyyy",
  "logo": {
    "name": "images/1Water.png",
    "width": 150,
    "height": 40
  },
  "bottomTable": {
    "freeFields": [
      {
        "name": "H/U",
        "isDate": false
      },
      {
        "name": "fromz",
        "isDate": false
      },
      {
        "name": "toz",
        "isDate": false
      }
    ]
  }
}
```

Figure 37 : Rapport du profil en long (Assainissement)

- ▶ **dateFormat** : format de la date du jour. Formats actuellement supportés : "dd/MM/yyyy", "dd-MM-yyyy" ;
- ▶ **logo** : définition du fichier de logo déposé dans le répertoire « \config\onewater\sanitation » :
  - **name** : nom du fichier avec éventuellement un ou plusieurs sous-répertoires ;
  - **width** : largeur en pixel ;
  - **height** : hauteur en pixel.
- ▶ **bottomTable** : Sous le rapport, un tableau permet de présenter des caractéristiques des tronçons du profil. Les 2 premières lignes sont réservées, et non modifiables, aux informations « Longueur » et « Pente ». L'administrateur peut définir 3 attributs supplémentaires à afficher en plus dans ce tableau avec la balise « freeFields » ;
- ▶ **freeFields** : champ défini par :

- **name** : nom de l'attribut dans la base de données. Si »name » contient « H/W » (voir l'exemple ci-dessus), la ligne contiendra la combinaison des attributs « Hauteur » et « Largeur » séparés par un « / » ;
- **isDate** : indique si l'attribut représente une date (true) ou pas (false).

### 3.2.7.3 MNT

La configuration du MNT est définie dans le groupe « mnt » :

```
"mnt": {  
    "densifyStep": 1,  
    "accessor": "gdal",  
    "source": "gdal001/morbihan.vrt"  
}
```

*Figure 38 : MNT pour profil en long (Assainissement)*

- **densifyStep** : la distance (mètres terrain) entre 2 mesures de hauteur ;
- **accessor** : indique le logiciel d'accès aux données du MNT (uniquement gdal dans cette version) ;
- **source** : indique le chemin du fichier « .vrt ». Si le chemin est relatif, il s'appuie sur le répertoire « <MonServeur>\config\permanent\mnt » ;

### 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 39 : 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) ;
- ▶ **fileTextDelimitior** : Délimiteur des chaines 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 introuvable 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 introuvable 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 40 : 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.

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.

- **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
- Etc...

```
"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 41 : 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'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).

► **edges**: Liste des linéaires concernés, définis par :

- **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.

### 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 42 : 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.

### 3.2.8.2 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 paramétrier 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 **α** utilisé dans le calcul de la note élémentaire d'une observation (**N=α<sup>p</sup>**). 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=α<sup>p</sup>**. 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 **α** 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 **α** 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  $\leq 1$  ;
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  $\leq 1$  ;
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  $\leq 1$  ;
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 « [tvi\\_1water\\_score\\_parameters.json](#) ».

Pour la notation RERAU le fichier JSON doit se nommer « [tvi\\_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 ;
  - ▶ 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 de trois façons différentes :

- ▶ **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 résultat déposés.

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 « **DirIn** ».

### 3.2.9 Intervention

La 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.

Water 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 43 : 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 44 : 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
- Etc...

```
"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 45 : 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_ACES",
            "combineUsingOr": true,
            "isSpecificValue": true
        }
    }
]
```

Figure 46 : 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:/IWATER_DEV/1DevSectorisation",
        "layerId": 1
    }
]
```

Figure 47 : 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 48 : 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'enchainement 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 49 : 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) :
    - o **name** : Nom de l'étape qui doit être unique ;
    - o **label** : Libellé de l'étape qui apparaît au niveau de la progression du prototype ;
    - o **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://1bz-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 50 : 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 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: <xMIRx5J19lnuoTO+0Pr2zx8N2YE7L2QPz1T+k03+Vo=
```

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

Exemple : `cadastre.database.password=~xMIRx5J19lnuoTO+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_config.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 valeur de l'attribut est toujours remplacée (valeur par défaut) ;
    - 1 : la valeur de l'attribut est remplacée si elle n'est pas renseignée ;
    - 2 : si la valeur de l'attribut est nulle elle est remplacée par la valeur constatée.

Exemple de fichier de configuration de gestion des conséquences :

```
{  
  "networkEntities": [  
    {  
      "domainNetworkName": "<Nom du domaine de réseau>",  
      "source": "<Nom de la source UN>",  
      "assetGroup": 123456,  
      "attributes": [  
        {  
          "name": "Attribut1",  
          "replaceMode": 0  
        },  
        {  
          "name": "Attribut2",  
          "replaceMode": 1  
        },  
        {  
          "name": "Attribut3",  
          "replaceMode": 2  
        }  
      ]  
    }  
  ]  
}
```

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

## 4 CONFIGURATION DE LA MOBILITE

### 4.1 FICHIERS DE CONFIGURATION

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 » ;

### 4.2 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 ».

#### 4.2.1 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": "<Décalage 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>  
}
```

##### 4.2.1.1 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 : Égale 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.

#### 4.2.1.2 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](#).

#### 4.2.1.3 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.

#### 4.2.1.4 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.

#### 4.2.1.5 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"  
}
```

#### 4.2.2 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  
    }  
  ]
```

```
]  
}
```

#### 4.2.2.1 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 ».

#### 4.2.2.2 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"
    }
  ]
}
```

#### 4.2.3 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
  }
}
```

#### 4.2.3.1 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.

#### 4.2.3.2 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"
      }
    ]
  }
}
```

## 4.3 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, est optionnel et 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).

## 4.4 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.

## 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 51 : 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"  
    },  

```

Figure 52 : 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 53 : 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 54 : 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 55 : 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 56 : 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 « 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": "%d",  
    "dictionary": "Dico_Diametre_Tronçon",  
    "next": [  
        {  
            "attribute": "diamext"  
        }  
    ],  
    "default": 100  
},
```

Figure 57 : 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 58 : 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 59 : Exemple de dictionnaire avec un opérateur de comparaison d'égalité implicite

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

Figure 60 : 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 « domaines ».

#### 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 [décrise 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 61 : 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 renournée par « attribut » ou « formule » de la définition de valeur parent.

La description de la liste de définitions de valeur est la même que celle [décrise 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 62 : 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 « attribut » 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 63 : 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 64 : 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 représentants 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 65 : 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 66 : 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 67 : 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 68 : 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 69 : 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 70 : 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 « 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": "%d",  
  "dictionary": "Dico_Diametre_Tronçon",  
  "next": [  
    {  
      "attribute": "diamext"  
    },  
    "default": 100  
  ],  
  "value": 100  
},
```

Figure 71 : 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  
  ],  
  "value": 100  
},
```

Figure 72 : 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": ">=",
        "value": "RECT_CLOSED"
      }
    ]
  }
],
```

Figure 73 : Exemple de dictionnaire avec un opérateur de comparaison d'égalité implicite

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

Figure 74 : 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 « domaines ».

#### 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 [décrise 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 75 : 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 renournée par « attribute » ou « formule » de la définition de valeur parent.

La description de la liste de définitions de valeur est la même que celle [décrise 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 76 : 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"
    }
  ],
  "next": []
},
```

Figure 77 : Partie « Paramètres »

## 5.2.2 Exemple

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

Figure 78 : 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 79 : 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'outil 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": {  
          "RegardNonVisitabile": {  
            "RegardVisitabile": {  
              ...  
            }  
          }  
        }  
      }  
    }  
  },
```

Figure 80 : 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  
      ],  
      "precisionXY": [  
        "Text",  
        1  
      ],  
      "precisionZ": [  
        "Text",  
        1  
      ],  
      "visibleS": [  
        "Short",  
        1  
      ],  
      "visibleE": [  
        "Short",  
        1  
      ]  
    }  
  },
```

Figure 81 : 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 82 : 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 83 : Exemple de dictionnaire avec un opérateur de comparaison d'égalité implicite

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

Figure 84 : 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 85 : 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 86 : 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": "visible",
    "ope": "==",
    "value": 1
  }
]

```

Figure 87 : 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 88 : 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 « domaines ».

#### 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 [décrise 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": "IW_CLASSE_PRECISION",
        "next": [],
        "default": 3
      },
      {
        "attribute": "qualitegeolocz",
        "format": "%s",
        "dictionary": "IW_CLASSE_PRECISION",
        "next": [],
        "default": 3
      }
    ]
  }
],
```

Figure 89 : 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 90 : 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 91 : Rechercher une valeur dans un dictionnaire**

## 5.4 MÉTHODE DE NOTATION

<sup>1</sup>Water 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-Troncon : 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 renvoyée 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 renvoyée 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 renvoyée 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 retournée 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 ne 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": "materiau",  
    "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
  - a. Si DATE\_REF < DATE\_POSE alors DATE\_INF = DATE\_POSE
  - b. 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 :
  - a. On calcule l'intersection entre l'équipement et la zone géographique,
  - b. On calcule SINTER = surface de l'intersection entre l'équipement et la zone géographique
  - c. 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 pas 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 partie 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 :
  - a. On calcule l'intersection entre l'équipement et la zone géographique,
  - b. Si l'intersection n'est pas vide Alors INDICE = valeur de l'attribut **level** de la zone
  - c. On calcule le SCORE à partir du tableau de conversion
  - d. On calcule SINTER = surface de l'intersection entre l'équipement et la zone géographique
  - e. SI  $(\text{SINTER} / \text{SEQU}) < \text{threshold}$  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					
		0	1	2	3	4	
SCORE NOTE N2	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 SELECTION PERSONNALISES

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 DEFINIR 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 symbolisation 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.