

MANUEL D'UTILISATION DE LA TECHNOLOGIE FDI-T® V4.0.2

Ce document décrit les fonctionnalités du logiciel FDI-T (*False Data Injection Testing*) pour les systèmes de contrôle aérien, ainsi que leurs différentes utilisations pour la génération et la concrétisation de scénarios d'attaques métiers.

La technologie FDI-T est développée par les sociétés Smartesting et Kereval.

Version du document	2.0
Version du logiciel	FDI-T 4.0.2
Etat	Validation
Date	24/04/2019

TABLE DES MATIÈRES

Éléments en sortie.....	4
1. Présentation de l'interface	5
2. Création	7
2.1. Création d'un projet.....	7
2.2. Création d'une zone	8
2.3. Création d'un scénario graphique ou textuel	9
2.4. Création d'une exécution	11
3. Éditeur graphique de scénarios.....	12
3.1. Fenêtre d'affichage des informations	13
3.2. Édition d'une action de scénario	14
3.2.1. Paramètres généraux	14
3.2.2. Édition d'une altération	15
3.2.3. Édition d'une saturation.....	17
3.2.4. Édition d'une suppression.....	18
3.2.5. Édition d'une action personnalisées [NON FONCTIONNEL en V4.0.2]	18
3.3. Édition des filtres d'un scénario	19
3.4. Édition d'assertions [NON FONCTIONNEL en V4.0.2]	20
3.4.1. Assertions globales	21
3.4.2. Assertions spécifiques [NON FONCTIONNEL en V4.0.2]	21
3.5. Affichage des plans de vol [NON FONCTIONNEL en V4.0.2]	22
4. Conception textuelle de scénarios	23
4.1. Éditeur de filtres	23
4.2. Éditeur de déclencheurs d'altération.....	25
4.3. Éditeur textuel de scénarios	28
5. Éditeur de zones.....	29
6. Éditeur de configurations d'exécutions	32

6.1.	Configuration serveur/client pour l'exécution de scénarios.....	32
6.1.1.	Partie « <i>Préparation</i> »	32
6.1.2.	La partie « <i>Serveur</i> »	34
6.1.3.	La partie « <i>Client</i> »	36
6.1.1.	La partie « Génération de rapport »	37
6.2.	Configuration d'un sensor de type SDPS	37
6.3.	Configuration d'un sensor modélisant un radar.....	38
6.4.	Configuration du serveur de logs [NON FONCTIONNEL en V4.0.2]	39
7.	Lancement CONVERTISSEUR sans GUI.....	39
7.1.	Conversion d'un fichier BEAST.....	40
7.1.1.	Conversion d'un fichier beast vers un fichier JSON	40
7.1.2.	Conversion d'un fichier JSON vers un fichier BEAST	40
7.1.3.	Conversion d'un fichier JSON vers des fichiers njson	40
7.2.	Conversion d'un fichier SBS	40
7.2.1.	Conversion d'un fichier SBS vers un fichier JSON	40
7.2.1.	Conversion d'un fichier JSON vers un fichier SBS	41
7.2.1.	Conversion d'un fichier JSON vers des fichiers njson	41
8.	lancement simulateur sans GUI	41
8.1.	Lancement du serveur	41
8.2.	Lancement du client.....	41
8.2.1.	Génération du rapport de test.....	42
8.2.1.1.	Génération du CSV.....	42
8.2.1.2.	Génération du rapport	42
9.	Problèmes connus et solutions	43
10.	Table des figures	45
	Annexe 1 : Spécification du langage dédié.....	47

ÉLÉMENTS EN SORTIE

Le répertoire d'installation du logiciel FDI-T en version 4.0.2 contient les éléments suivants :

- L'exécutable de l'application
- La documentation
- Une notice d'utilisation
- Un fichier de configuration serveur : *sensorConfig.xml*
- Un fichier de configuration client : *testEvents.xml*
- Un répertoire de travail nommé « workspace » : exemple d'un projet

PRÉREQUIS

Version de Java à utiliser : **IL FAUT IMPÉRATIVEMENT INSTALLER UNE VERSION DE JAVA SUPÉRIEURE OU ÉGALE À LA VERSION 1.8_092 (update 092) POUR POUVOIR FAIRE FONCTIONNER LE SIMULATEUR.**

1. PRESENTATION DE L'INTERFACE

L'interface de l'application FDI-T se décompose en deux parties :

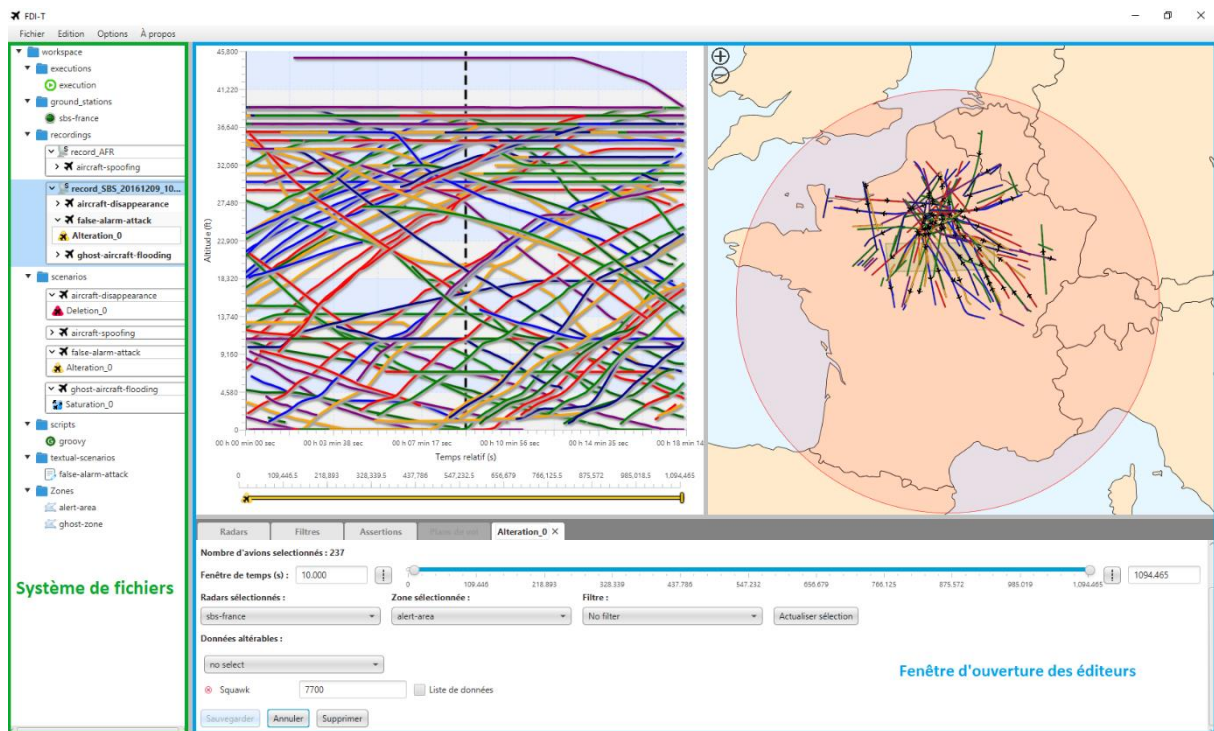


Figure 1 : Interface graphique de FDI-T






1. Le système de fichiers (encadré vert sur la figure 1) : Il est possible d'ajouter, modifier ou supprimer les fichiers pouvant être traités par l'application uniquement.
2. Fenêtre d'ouverture des éditeurs (encadré bleu sur la figure 1), il existe sept types d'éditeur dont les fonctionnements sont décrits dans ce document :
 - Éditeur graphique de scénarios
 - Erreur ! Source du renvoi introuvable.
 - Erreur ! Source du renvoi introuvable.
 - Erreur ! Source du renvoi introuvable.
 - Éditeur de radars
 - Éditeur de zones
 - Erreur ! Source du renvoi introuvable.

Il existe différentes icônes pour les différents fichiers chargés dans l'application. Les icônes suivantes sont cliquables et ont pour effet d'ouvrir l'éditeur qui correspond au type de fichier qu'elles représentent :

 → Un répertoire.




 → Fichier .gscenario : fichier de scénario graphique.

 → Fichier .scenario : fichier de scénario textuel.

-  → Fichier .zone : fichier de zone géographique.
-  → Fichier .xml : fichier de configurations d'exécutions.
-  → Fichier .sensor : fichier de configuration de radar.
-  → Fichier .trg : fichier de déclencheur d'action de scénario.
-  → Fichier .lfl : fichier de filtre.

La création des éléments cliquables est réalisée depuis un menu contextuel, en effectuant un clic droit sur le dossier sélectionné, sauf pour les enregistrements, qui ne peuvent pas être ouverts dans un éditeur.

Les autres icônes ne sont pas cliquables, elles indiquent juste la présence de fichier reconnu par FDI-T, et utilisable dans des autres éditeurs :

-  → Fichier .sbs/.bst : fichier d'enregistrement. Ce type de fichier est utilisé à la création d'un scénario.
-  → Fichier .fpl : fichier de plans de vol. Il peut être lié à un scénario existant depuis l'éditeur graphique de scénarios.
-  → Fichier .gy : fichier de script Groovy. Il peut être lié à des assertions globales depuis l'éditeur graphique de scénarios.

Les éléments non cliquables ne peuvent pas être créés depuis l'interface graphique, ils doivent être importés grâce à la fonctionnalité prévue à cet effet « *Fichier > Importer (Ctrl+I)* », ou être présents dans le répertoire de travail lors de son ouverture dans l'application.

2. CREATION

2.1. Création d'un projet

Un projet est un dossier qui contiendra tous les fichiers nécessaires pour la génération et la concrétisation des scénarios d'attaques.

Pour commencer, depuis le système d'exploitation, créer un dossier vide ayant le nom du projet. Dans ce dossier, il faut obligatoirement y ajouter un fichier d'enregistrement, nécessaire pour la création d'un scénario.

Il est possible de structurer le projet pendant cette étape. Ci-dessous, un exemple de structure pour un projet :

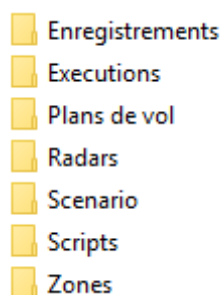


Figure 2 : Exemple de structure d'un projet

Ici, le dossier racine du projet est *Demo*. Dans ce dossier se trouvent cinq autres dossiers :

- *Enregistrements* : dossier contenant l'ensemble des enregistrements du projet.
- *Executions* : dossier contenant l'ensemble des fichiers de configurations d'exécution, avec les deux fichiers de configurations : *sensorConfig.xml* et *testEvents.xml*
- *Scenario* : dossier contenant l'ensemble des scénarios du projet.
- *Zones* : dossier contenant l'ensemble des zones géographiques créées.
- *Radars* : dossier contenant l'ensemble des radars créés.
- *Scripts* : dossier contenant l'ensemble des scripts Groovy du projet.
- *Plans de vol* : dossier contenant l'ensemble des fichiers de plans de vol du projet.

Ensuite, lancer l'application et ouvrir le dossier du projet depuis la barre de menu « *Fichier > Ouvrir (Ctrl+O)* » :

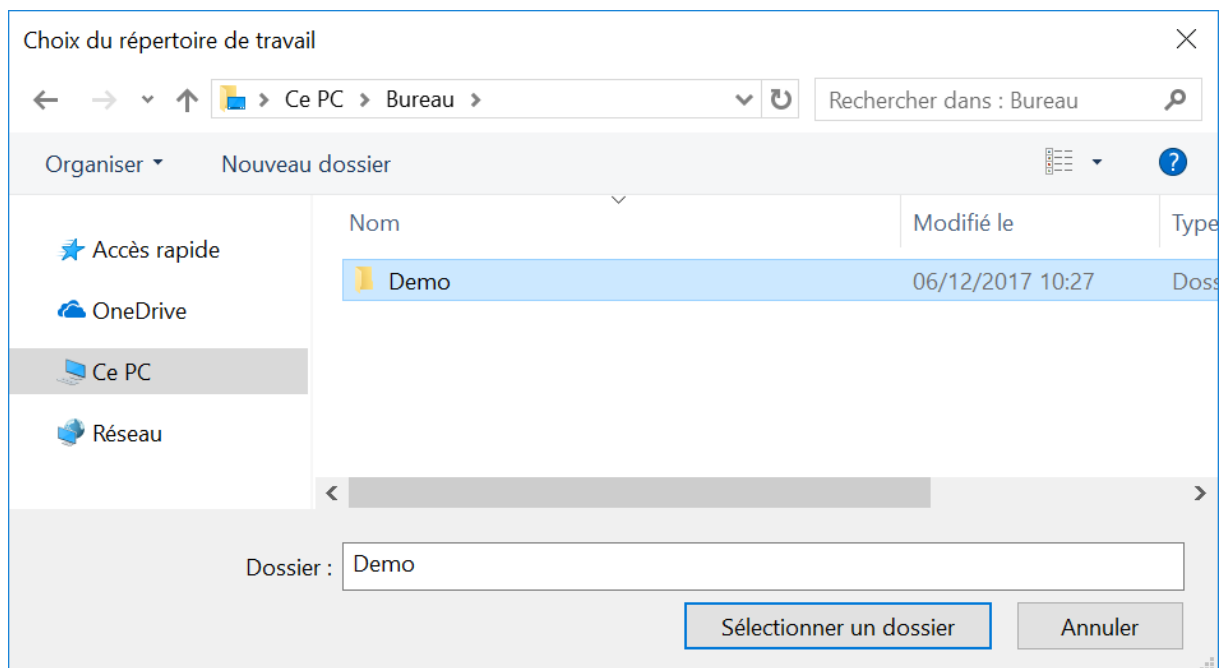


Figure 3 : Système de fichiers - Fenêtre d'ouverture d'un projet

Le dossier est ensuite chargé dans le système de fichiers :

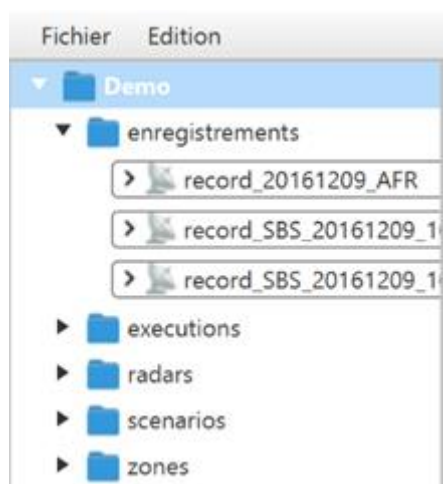


Figure 4 : Système de fichiers - Affichage d'un projet

2.2. Création d'une zone

Pour créer une zone, effectuer sur un dossier un clic droit « *Créer > Zone* », une fenêtre de création s'ouvre :

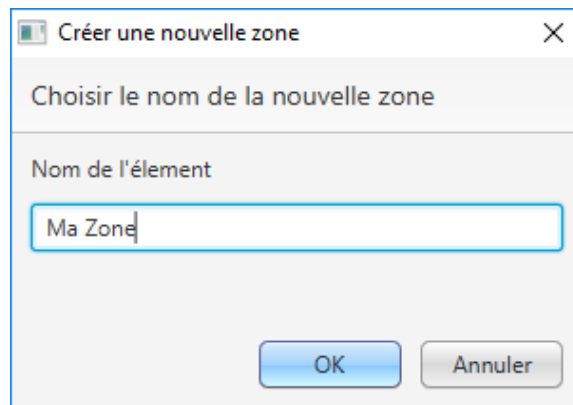


Figure 5 : Editeur de zones – Fenêtre de création d’une zone

Pour créer une zone, il faut choisir un nom de zone qui n’est **pas déjà utilisé par un autre élément du projet**, puis cliquer sur « OK ». L’Éditeur de zones s’ouvre alors, voici la représentation d’une zone existante dans le système de fichier :

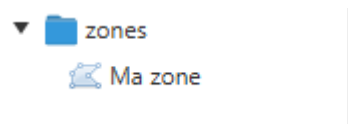


Figure 6 : Système de fichiers - Affichage d'une zone

Effectuer un double-clic sur l’icône ouvre la zone dans l’Éditeur de zones

2.3. Création d’un scénario graphique ou textuel

Effectuer un clic droit sur un dossier :

- « Créer > Scénario graphique » permet de créer un scénario graphique.
- « Créer > Scénario textuel » permet de créer un scénario textuel.

Dans les deux cas, la même fenêtre de création s’ouvre :

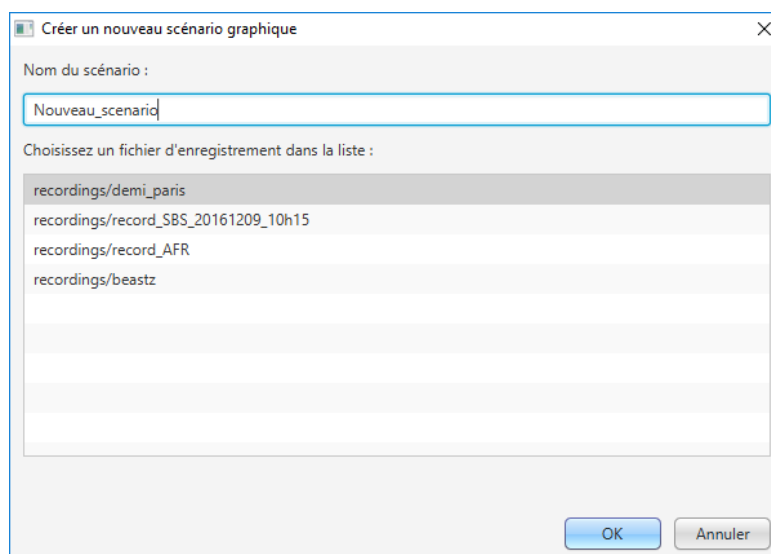


Figure 7 : Fenêtre de création d'un scénario

Pour créer un scénario, il faut lui donner un **nom unique dans le projet**, et lui associer un enregistrement. Cliquer sur « OK », puis l'éditeur graphique de scénario s'ouvre ([Éditeur graphique de scénarios](#)) sur l'enregistrement associé au scénario. Cette opération peut prendre du temps suivant la taille de l'enregistrement.

Dans le cas d'un scénario textuel, la fenêtre de création classique s'ouvre, seul **un nom unique** est requis pour la création d'un scénario textuel qui ne nécessite pas l'association d'un enregistrement.

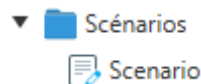


Figure 8 : Système de fichier - Affichage d'un scénario textuel

Effectuer un double-clic sur l'icône ouvre le scénario dans [l'éditeur de scénario textuel](#).

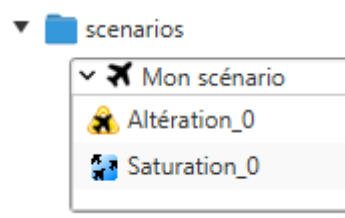


Figure 9 : Système de fichiers - Affichage d'un scénario

L'aspect d'un scénario graphique ou d'un enregistrement dans le système de fichiers est différent de celui des autres types de fichiers pouvant être chargés. Effectuer un double-clic sur le nom du scénario ouvre [l'Éditeur graphique de scénarios](#).

La liste déroulante en dessous du scénario contient l'ensemble des attaques métiers du scénario. Effectuer un double-clic l'ouvre dans l'éditeur d'altérations métiers de [l'Éditeur graphique de scénarios](#).

Le scénario graphique et les altérations métiers sont également accessibles depuis l'icône de l'enregistrement auquel le scénario est rattaché, via le même système de liste déroulante que pour les icônes de scénario, en voici un exemple :

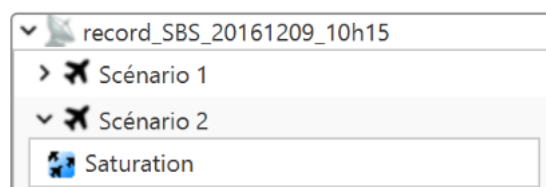


Figure 10 : Affichage d'un enregistrement et de ses scénarios rattachés

Les scénarios de la liste déroulante sous un enregistrement sont simplement des raccourcis vers les vrais fichiers scénario, il n'est donc pas possible de les déplacer, les supprimer ou les

renommer à partir de ces raccourcis. Pour cela, il faudra accéder aux vrais fichiers tels qu'ils sont présentés dans la Figure 9.

2.4. Création d'une exécution

Pour créer une configuration, effectuer sur un dossier un clic droit *Créer > Exécution*, une fenêtre de création s'ouvre :

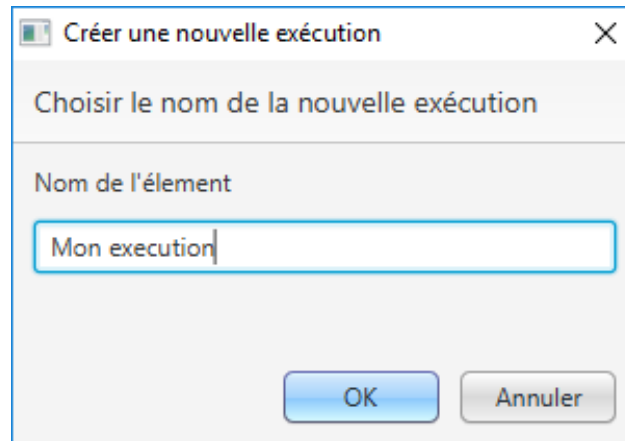


Figure 11: Éditeur de configurations d'exécutions - Fenêtre de création d'une configuration

Pour créer une configuration, il faut lui donner un nom unique dans le dossier de création de la configuration. Cliquer sur « OK », puis [l'Éditeur de configurations d'exécutions](#) s'ouvre. Voici la représentation d'une configuration existante dans le système de fichiers :

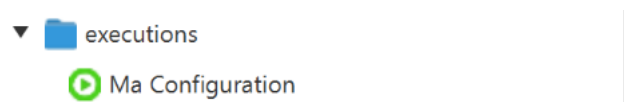


Figure 12 : Système de fichiers -Affichage d'une configuration d'exécution

Effectuer un double-clic ouvre la configuration dans [l'Éditeur de configurations d'exécutions](#).

3. ÉDITEUR GRAPHIQUE DE SCENARIOS

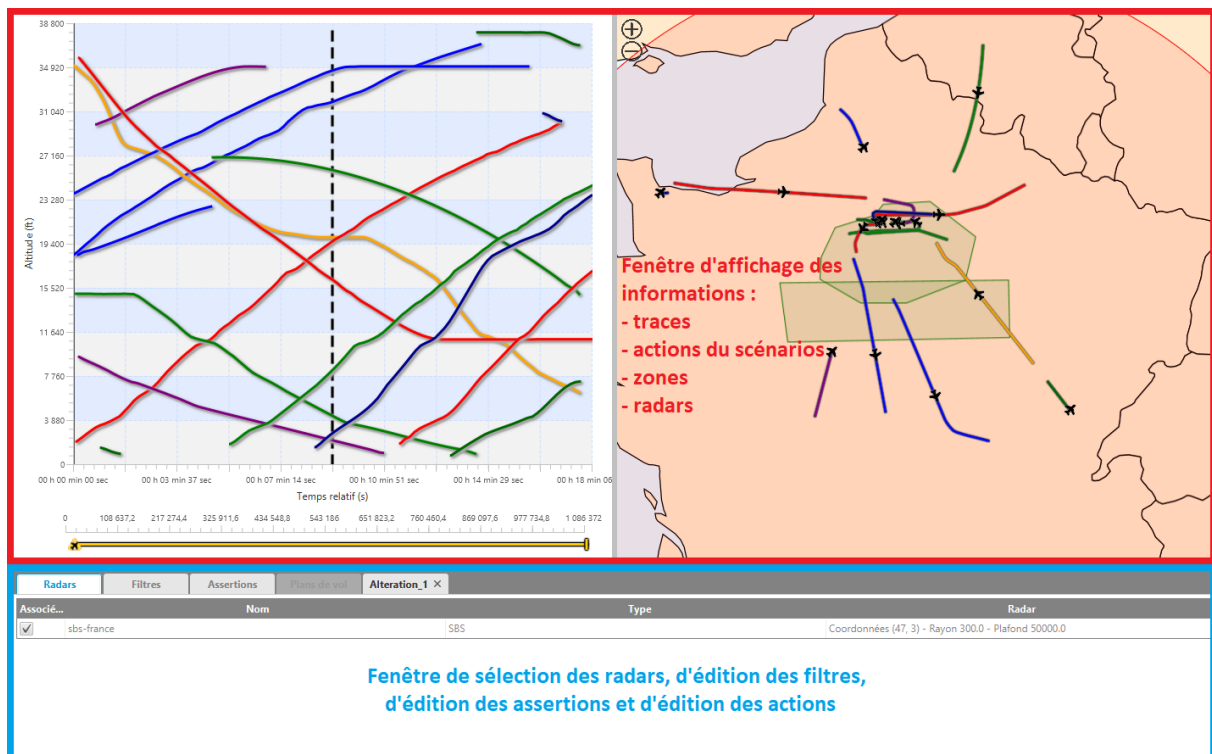


Figure 13 : Éditeur graphique de scénarios

L'éditeur est divisé en deux parties :

- Une fenêtre pour afficher les différentes informations pour la création des attaques métiers (encadré rouge).
- Une fenêtre d'édition depuis laquelle il est possible d'éditer les filtres afin d'extraire certaines pistes d'un enregistrement, de sélectionner les radars ou d'éditer les attaques métiers (encadré bleu).

3.1. Fenêtre d'affichage des informations

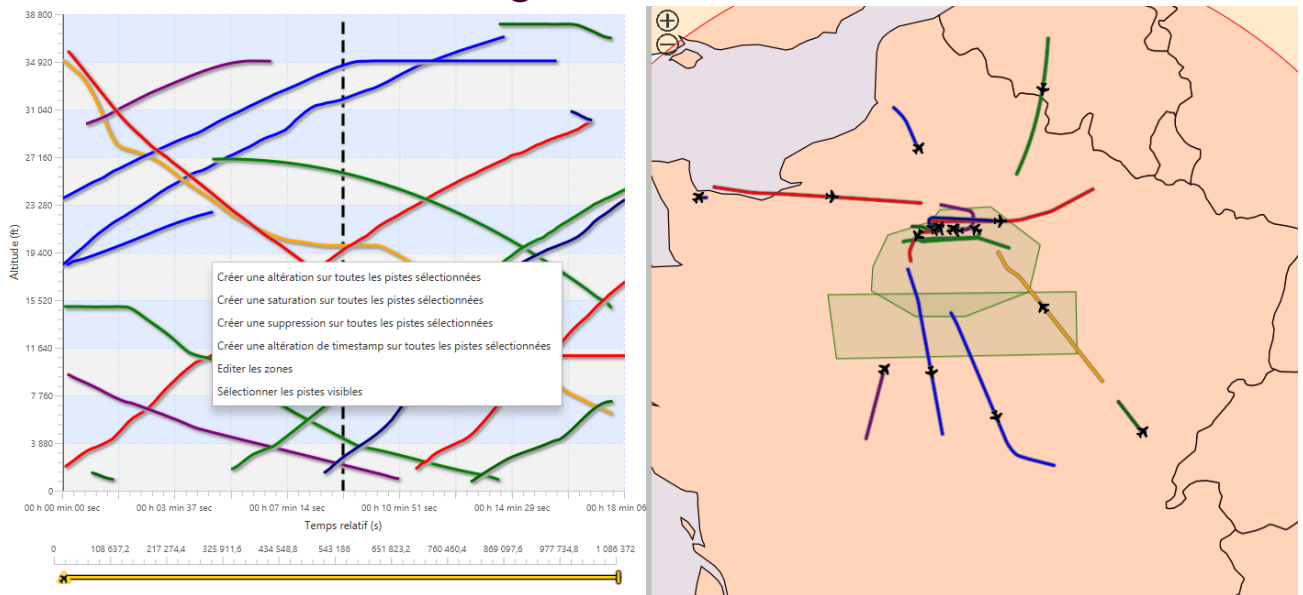


Figure 14 : Éditeur graphique de scénarios - Fenêtre d'information

La fenêtre affichant les informations possède deux parties :

- Dans celle de gauche, un graphique montrant l'évolution de l'altitude des pistes en fonction du temps, ainsi que sous ce graphique, les altérations métiers présentes dans le scénario.
- Dans celle de droite, les trajectoires des pistes sur une carte ainsi que les zones créées et les radars associés au scénario.

Il est possible d'interagir sur ces éléments via la souris. Dans la fenêtre de gauche, un clic droit fait apparaître un menu pour « *Créer une altération sur toutes les pistes sélectionnées* » ou encore accéder à [l'Éditeur de zones](#). Un clic simple sur une trace permet de la sélectionner ou la désélectionner.

Double-cliquer sur une attaque métier l'ouvre dans un onglet de la fenêtre d'édition. L'icône de l'attaque métier est différente selon le type de scénario visé :

- 🛡️ → Icône d'une altération de données.
- 🔒 → Icône d'une saturation.
- 💣 → Icône d'une suppression de piste.
- ⚙️ → Icône d'une action personnalisée.
- 🕒 → Icône d'une altération de timestamp
- 📡 → Icône d'une altération de *downlink format*

Enfin, déplacer le curseur noir sur le graphique modifie la position des avions sur la carte.

3.2. Édition d'une action de scénario

Pour ouvrir l'éditeur, il faut soit [créer l'action](#) ou demander l'édition depuis [la fenêtre d'information](#) ou depuis [le système de fichier](#). Les actions possèdent deux types de paramètres :

- Les paramètres généraux, qui sont communs à tous les types d'action.
- Les paramètres spécifiques, qui ne seront pas les mêmes selon le type d'action.

3.2.1. PARAMETRES GENERAUX

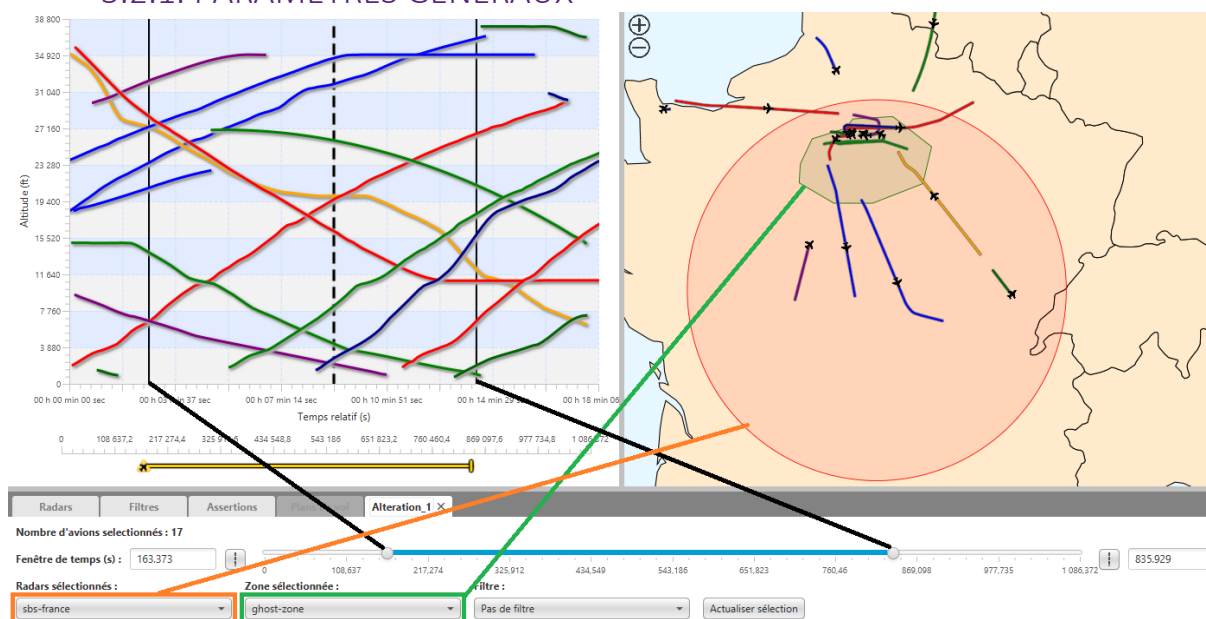


Figure 15 : Éditeur graphique de scénarios - Édition d'une action : paramètres généraux

Les paramètres d'action visibles dans la Figure 15 ne dépendent pas du type de l'action en cours d'édition, à partir de l'onglet d'édition de n'importe quel type d'action, il est possible de :

- Changer le nom de l'action avec un double-clic sur l'onglet correspondant à cette action.
- Changer la date de début et de fin de l'action.
- Choisir une zone dans laquelle se déroulera l'action.
- Choisir un filtre.

Certains changements sont visibles directement dans la fenêtre du dessus tels que le choix de la zone ou le changement de la fenêtre de temps.

Les paramètres spécifiques au type de l'action, sont présentés dans les sous-parties suivantes. Au niveau de l'IHM, ils apparaissent directement à la suite de la Figure 15.

Le bouton « *Actualiser sélection* » permet de valider un filtre après qu’il ait été édité. En effet, cette édition ne modifie pas l’affichage des pistes des attaques en cours d’édition.

3.2.2. ÉDITION D’UNE ALTERATION

Données altérables :

no select

✖

Squawk

7700

☐ Liste de données

✖

Altitude (ft)

25000

☐ Plage de données ☐ With offset

Sauvegarder

Annuler

Supprimer

Figure 16 : Éditeur graphique de scénario – Paramètres spécifiques d’une altération

Les paramètres spécifiques d’une altération sont visibles en Figure 16. Il est possible de choisir un attribut via une liste déroulante, et de fixer la valeur à laquelle il doit être altéré. Lors de l’altération des attributs, il est possible de :

- Fixer la valeur d’un attribut.
- Ajouter une valeur à un attribut décimal.
- Spécifier une liste de valeurs à prendre pour un attribut hexadécimal.
- Spécifier un intervalle de valeurs à prendre pour un attribut décimal.

Altération	Description
Squawk : 7700	Fixe le Squawk envoyé par l’avion à 7700
Altitude : 1000 Sans décalage	Fixe l’altitude à 1000ft pour l’ensemble des positions connues de l’avion
Altitude : 1000 Avec décalage	Augmente l’altitude de 1000ft pour l’ensemble des positions connues de l’avion
Altitude : 2000 – 3000 Sans décalage	Pour chaque valeur aux bornes de l’intervalle, crée un scénario qui fixe l’altitude à cette valeur.
Altitude 2000 – 3000 Avec décalage	Pour chaque valeur aux bornes de l’intervalle, crée un scénario qui augmente l’altitude de cette valeur.
Squaw : 7700 ; 7701 ; 7702 ; 7703	Pour chaque valeur de la liste, crée un scénario qui fixe le Squawk envoyé par l’avion à cette valeur.

Figure 17 : Éditeur graphique de scénarios - Exemple d'altération sur les attributs

Enfin l’attribut ICAO peut prendre la valeur RANDOM. Ainsi lors de la concrétisation, les ICAO des pistes altérées seront générés aléatoirement.

Altération				
Champs altéré	Valeur	Avec décalage ?	RANDOM	Massification
Altitude	Entier	Oui	Non	Intervalle
CallSign	8 caractères maximums (0-9 A-Z)	Non	Non	Liste
Emergency (Urgence)	0 ou 1	Non	Non	Aucune
GroundSpeed (Vitesse au sol)	Décimal	Oui	Non	Intervalle
ICAO	Entier hexadécimal (0-9 A-F) avec 6 chiffres	Non	Oui	Liste
Latitude	Décimal	Oui	Non	Intervalle
Longitude	Décimal	Oui	Non	Intervalle
SPI	0 ou 1	Non	Non	Aucune
Squawk	Entier avec 4 chiffres	Non	Non	Liste
Track (Route)	Décimal	Oui	Non	Aucune

Figure 18 : Éditeur graphique de scénarios - Tableau récapitulatif des altérations possibles

Une altération peut être massifiée, c'est-à-dire qu'un seul attribut peut prendre plusieurs valeurs afin de générer plusieurs scénarios à l'exécution. Il y a deux sortes de massification : par listes et par intervalles de valeurs.

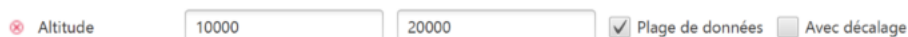


Figure 19 : Massification par intervalle de valeurs

Pour massifier à partir d'un intervalle de valeurs, il faut cocher la case *Plage de données*, ce qui a pour effet de faire apparaître un nouveau champ de saisie. Le premier champ représentant la borne inférieure de l'intervalle et le second champ la borne supérieure.

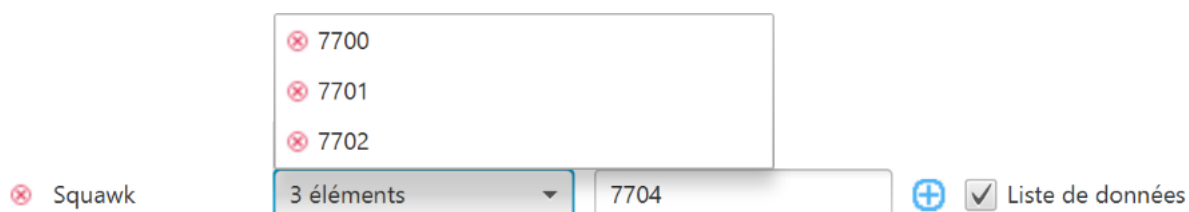


Figure 20 : Massification par liste de valeurs

Pour massifier à partir d'une liste de valeurs, il faut cocher la case *Liste de données*, ce qui a pour effet de faire apparaître des éléments graphiques permettant d'agrémenter une liste avec un ensemble de valeur. Pour ajouter une valeur, il faut remplir le champ de saisie puis cliquer sur le bouton « + », la valeur ira dans la liste déroulante. Pour supprimer cet élément de la liste, il suffit de le sélectionner dans cette dernière.

Au cours de l'exécution la massification a pour effet de créer une multitude de scénarios. Pour chaque attribut massifié à partir d'un intervalle de valeurs, cinq scénarios seront créés fixant chacun l'attribut aux valeurs suivantes :

- Borne inférieure + 1
- Borne inférieure- 1
- Valeur médiane
- Borne supérieure + 1
- Borne supérieure - 1


Pour les attributs massifiés à partir de liste, le principe est le même, sauf que dans chacun des scénarios, la valeur de l'attribut sera fixée à une valeur de la liste. Dans le cas où plusieurs attributs sont massifiés, un produit cartésien entre toutes les valeurs possibles est alors effectué pour créer les scénarios. Voici un exemple :

Soit le scénario suivant : les **altitudes** de tous les appareils sont altérées suivant l'intervalle de valeur **10000 à 20000** et leurs **Squawk** sont altérés suivant la liste à deux éléments : **7700 et 7701**.

Ces deux massifications vont produire les cas de test présentés dans le tableau suivant :

Cas de test	Valeurs des attributs
1	Altitude = 9999 ; Squawk = 7700
2	Altitude = 9999 ; Squawk = 7701
3	Altitude = 10001 ; Squawk = 7700
4	Altitude = 10001 ; Squawk = 7701
5	Altitude = 15000 ; Squawk = 7700
6	Altitude = 15000 ; Squawk = 7701
7	Altitude = 19999 ; Squawk = 7700
8	Altitude = 19999 ; Squawk = 7701
9	Altitude = 20001 ; Squawk = 7700
10	Altitude = 20001 ; Squawk = 7701

3.2.3. ÉDITION D'UNE SATURATION

Nombre d'avions pour la saturation : 

Données altérables :

☒ ICAO ☐ Liste de données

Figure 21 : Éditeur graphique de scénarios – Paramètres spécifiques d'une saturation

Une saturation possède deux paramètres spécifiques :

- Le champ « *Nombre d'avions pour la saturation* » qui permet de spécifier le nombre d'avions qu'il y aura autour des avions sélectionnés.
- L'attribut ICAO, qui est le seul attribut sélectionnable dans la liste déroulante.

3.2.4. ÉDITION D'UNE SUPPRESSION

Une action de suppression de piste ne possède aucun paramètre spécifique, car les paramètres généraux suffisent à sélectionner les pistes à supprimer.

3.2.5. ÉDITION D'UNE ACTION PERSONNALISEES [NON FONCTIONNEL EN V4.0.2]



Classe :

MySaturation

Paramètres :

Ajouter un paramètre

⊗ Clé :	number	Valeur :	10
---------	--------	----------	----

Sauvegarder Annuler Supprimer

Figure 22 - Éditeur graphique de scénario – Paramètres spécifiques d'une action personnalisée

Les paramètres spécifiques d'une action personnalisée sont visibles en Figure 22, il y en a deux :

- Le paramètre « *Classe* » permet, à partir d'une liste déroulante, de choisir la classe de l'action personnalisée. Cette classe se base sur une librairie externe à FDI-T et personnalisable par l'utilisateur afin de créer ses propres altérations.
- Le deuxième paramètre est une liste de pairs « Clé », « Valeur ». Cette liste permet à l'utilisateur de faire passer des paramètres aux altérations qu'il a définies dans la librairie externe.

Pour les quatre types d'actions, on remarque dans les figures : Figure 16, Figure 21 et Figure 22 la présence de trois boutons :

- Le bouton « *Sauvegarder* » enregistre les changements. Tant que les changements ne sont pas sauvegardés, ils ne sont pas effectifs dans les autres éditeurs.
- Le bouton « *Annuler* » annule tous les changements effectués depuis la dernière sauvegarde.
- Le bouton « *Supprimer* » supprime l'action.

3.3.Édition des filtres d'un scénario

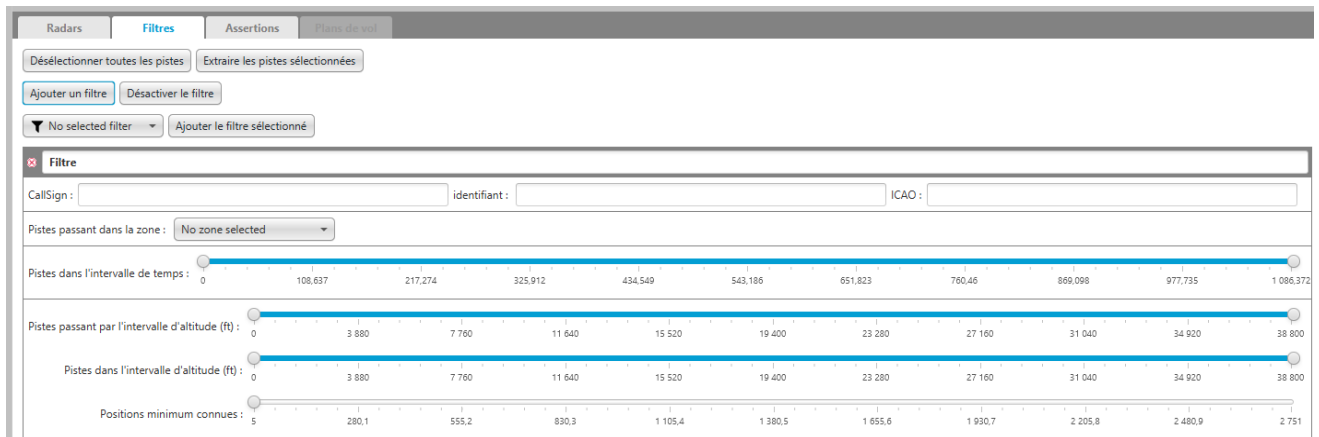


Figure 23 : Éditeur graphique de scénarios – Onglet d'édition des filtres

Dans l'onglet « *Filtres* » il est possible d'effectuer les actions suivantes :

- Le bouton « *Sélectionner toutes les pistes* » permet de sélectionner toutes les pistes **affichées** dans la fenêtre d'information de l'éditeur.
- Le bouton « *Désélectionner toutes les pistes* » permet de désélectionner toutes les pistes.
- Le bouton « *Extraire les pistes sélectionnées* » permet de créer un nouveau fichier d'enregistrement, en extrayant les pistes sélectionnées. Le fichier d'enregistrement créé est placé dans le même dossier que l'enregistrement original. Le nom par défaut du fichier est : *nomDuFichierOriginal_extractX*.
- Le bouton « *Ajouter un filtre* » ajoute un nouveau filtre à la liste des filtres créés. Ce filtre possède les conditions par défaut pour afficher toutes les traces pouvant être dessinées.
- Le bouton « *Désactiver le filtre* » désactive le filtre courant (visuellement, le cadre d'édition de tous les filtres est fermé).
 - Le bouton « *Ajouter le filtre sélectionné* » ajoute le filtre textuel sélectionné dans la liste déroulante à la liste des filtres créés. Les filtres textuels peuvent être créés et édités depuis l'[Erreur ! Source du renvoi introuvable.](#)

Cliquer sur le nom d'un des filtres de la liste active le filtre, et ouvre le cadre d'édition de celui-ci. Dans la Figure 23, le filtre « *Filtre* » est actif.

Les actions possibles pour éditer un filtre sont :

- Nommer le filtre
- Filtrer selon les attributs d'un avion : CallSign & ICAO, ils contiennent le pattern saisi. Pour l'identifiant, il commence par le pattern saisi.

- Filtrer en fonction d'une zone géographique, en gardant les traces passant dans la zone sélectionnée.
- Filtrer en fonction d'un intervalle de temps
- Filtrer en fonction d'un intervalle borné par une altitude. Pour que les pistes soient incluses dans l'intervalle ou passent dans l'intervalle.
- Filtre en fonction du nombre de positions valides connues dans l'enregistrement.

Il est possible aussi de combiner ces éléments.

Pour le CallSign & ICAO → le caractère '*' est un caractère spécial qui correspond à un ou plusieurs caractères.

Exemples de filtres sur les attributs :

Filtre	Description	Exemples valides
CallSign : A	Filtre l'ensemble des avions avec un CallSign <u>contenant la lettre A</u>	CallSign = KIRALA CallSign = ALLTHR
CallSign : *	Filtre l'ensemble des avions avec un CallSign <u>contenant au moins un caractère</u> , c'est-à-dire l'ensemble des avions ayant un CallSign connu.	CallSign = A CallSign = OKLMMDR
Identifiant : 1	Filtre l'ensemble des avions avec un Identifiant <u>commençant</u> par 1	Identifiant = 1010 Identifiant = 170 ...
ICAO : Z*A	Filtre l'ensemble des avions avec un ICAO <u>contenant Z puis A</u>	ICAO = ZASTRA ICAO = YZERA
CallSign : AFR Identifiant : 170	Filtre l'ensemble des avions avec un CallSign <u>contenant AFR</u> et un identifiant <u>commençant par 170</u>	CallSign = AFR817 et Identifiant = 17089 CallSign = 12AFR12 et Identifiant = 170170

Figure 24 : Éditeur graphique de scénarios - Exemples de filtres sur les attributs

Lors de l'édition des filtres, les traces ne correspondant pas au filtre courant deviennent invisibles dans la partie information. De plus, toutes les pistes invisibles redevenant visibles sont désélectionnées.

L'édition d'un filtre ne change pas la sélection des pistes d'une altération en cours d'édition utilisant ce filtre, il faut accepter les changements dans l'onglet d'édition de l'altération.

3.4.Édition d'assertions [NON FONCTIONNEL en V4.0.2]

Une assertion représente un résultat attendu, et plus précisément un texte qui sera recherché dans des retours syslog du système sous test.

3.4.1. ASSERTIONS GLOBALES

Il est possible de définir des assertions globales, qui seront évaluées pendant un laps de temps données. Elles se définissent depuis l'éditeur graphique de scénarios, dans l'onglet « Assertion ».

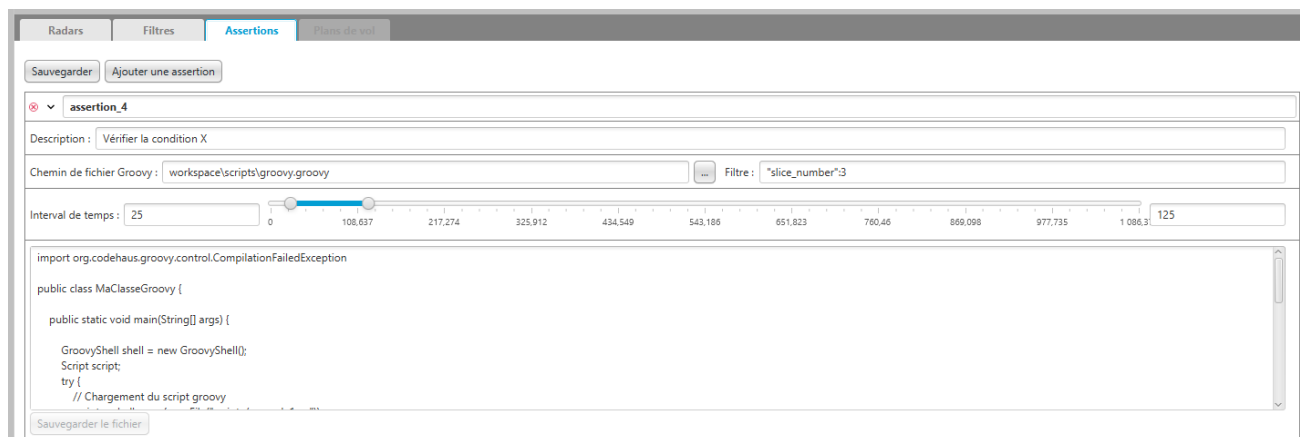


Figure 25 - Exemple d'assertion globale

Les assertions se basent sur un fichier de script Groovy (.gy), pour être utilisé dans une assertion le script doit se trouver dans l'espace de travail de FDI-T lors de son ouverture ou être importer via la fonctionnalité dédiée. Le contenu du fichier peut être modifié directement depuis l'éditeur d'assertions.


Le bouton « *Sauvegarder le fichier* » permet d'écrire le contenu de la zone de texte de la Figure 25 dans le fichier Groovy sélectionné.

La création d'une assertion se fait avec le bouton « *Ajouter une assertion* »

3.4.2. ASSERTIONS SPECIFIQUES [NON FONCTIONNEL EN V4.0.2]

Il est également possible de définir une assertion pour un incident donné. Les assertions, se définissent dans le champ « Assertion » prévu pour :

Nombre d'avions sélectionnés : 158

Fenêtre de temps (s) : 

Zone sélectionnée : Filtre :

Données altérables :

☒ Squawk ☐ Liste de données

Figure 26 : Définition d'une assertion

3.5. Affichage des plans de vol [NON FONCTIONNEL en V4.0.2]

Dans l'éditeur graphique de scénario, l'onglet « *Plans de vol* » permet d'associer un fichier de plans de vol à un scénario.

Radars	Filtres	Assertions	Plans de vol
Choisir les plans de vol			
▼ Callsign : NJE937L	Départ: LFPB	Arrivé: LSGG	301 mi.
Type d'aéronef : C68A	Heure de départ : 2018-02-20T13:15	Transmission en vol : <input type="checkbox"/>	
> Callsign : SAS834	Départ: LFPG	Arrivé: ENGM	759 mi.
> Callsign : AFR71EX	Départ: LFPG	Arrivé: EPWA	787 mi.
> Callsign : AFR79FH	Départ: LFPG	Arrivé: EGBB	302 mi.
> Callsign : AAF819	Départ: LFPO	Arrivé: EDDT	515 mi.
> Callsign : AFR968	Départ: LFPG	Arrivé: MDPC	3905 mi.
> Callsign : N552AV	Départ: KUGN	Arrivé: LFPB	3626 mi.
> Callsign : EZY21LU	Départ: LFPG	Arrivé: EGPF	505 mi.

Figure 27 - Éditeur graphique de scénarios – Affichage des plans de vol

La Figure 27 montre la manière dont sont présentés les plans de vol. Chaque ligne affiche les informations importantes d'un plan de vol (callsign, aéroport de départ, aéroport d'arrivée, distance en miles). Une ligne peut être dépliée afin d'afficher des informations supplémentaires.



Figure 28 - Éditeur graphique de scénario - Choix d'un fichier de plans de vol

L'association d'un fichier de plans de vol à l'enregistrement d'un scénario graphique avec le bouton « *Choisir les plans de vol* ». Ce bouton a pour effet d'ouvrir une boîte de dialogue permettant de choisir un fichier de plans de vol parmi ceux présents dans l'espace de travail.

Pour être sélectionné, un fichier de plans de vol devait donc se trouver dans l'espace de travail à l'ouverture de ce dernier, ou avoir été importé par l'utilisateur.

4. CONCEPTION TEXTUELLE DE SCENARIOS

L'éditeur textuel de scénario permet à un utilisateur de créer un scénario à partir d'un langage dédié. Tout en restant intuitive, cette alternative à la création graphique de scénarios se veut générique, c'est-à-dire qu'un scénario est applicable à n'importe quel enregistrement. De plus, l'approche textuelle offre un pouvoir d'expression accru, rendant ainsi possible la définition de scénarios complexes, précis et ciblés.

Le processus de conception textuelle de scénario se compose de trois étapes :

1. Définition des avions impliqués dans le scénario
2. Définition des plages temporelles d'altération
3. Définition du corps du scénario

Chacune de ces étapes est effectuée au moyen d'un éditeur dédié et d'un langage associé. Les deux premières étapes, le filtrage d'avion et la création de plages temporelles d'altération, s'appuient sur l'expression de conditions booléennes portant sur les propriétés des avions, que ce soit vis-à-vis de tout l'enregistrement, pour le filtrage, ou à un instant donné, pour la définition des plages temporelles d'altération.

4.1. Éditeur de filtres

Il est très fréquent qu'un scénario ne cible qu'une sous partie des avions observés lors d'un enregistrement. Ce ciblage est parfois trivial, comme par exemple « *seulement les avions de la compagnie Air-France* ». Dans d'autres cas cependant, le ciblage requis peut s'avérer complexe et précis, par exemple « *seulement les avions dont l'altitude n'a jamais excédé 30000ft et qui se sont trouvés au moins une fois dans la zone **ZRT*** ». Un éditeur dédié de filtre (génériques) est ainsi proposé pour rendre possible de tel ciblage.

L'expression de filtre s'appuie sur un langage d'expression booléen. Il est ainsi possible d'exprimer des conditions liées aux propriétés intrinsèques des avions, comme l'altitude, la vitesse au sol, le callsign, etc. Il est de plus possible de combiner des conditions avec les opérateurs logiques « *ET* » et « *OU* », permettant respectivement la conjonction et la disjonction de conditions logiques. Enfin, et du fait de la nature dynamique de beaucoup de propriétés d'avion comme l'altitude, coordonnées ou la vitesse au sol, deux opérateurs temporels issus de la logique LTL ont été introduits au langage de filtre :

- **G** (expr) : L'expression *expr* doit être vraie pendant tout l'enregistrement.
- **F** (expr) : L'expression *expr* doit être vraie au moins une fois au cours de l'enregistrement.

Ainsi, les conditions exprimées portent sur l'ensemble de l'enregistrement. A noter que les opérateurs temporels ne sont pas applicables aux expressions incluant des propriétés statiques, comme l'ICAO par exemple.

L'éditeur de filtre est présenté en Figure 29. La partie centrale permet l'écriture de conditions, aidée d'un mécanisme de coloration syntaxique, d'auto-complétion et d'aide à la correction des erreurs afin d'assurer une utilisabilité au moins équivalente aux éditeurs de langage de programmation modernes. Le filtre décrit en Figure 29 est comme suit :

```
eval not(G(ALTITUDE > 30000)) and F(ALTITUDE > 34000 and LONGITUDE < 2.30)
```

Le mot clé **eval** détermine le début d'une expression de filtrage. L'expression qui suit est une conjonction de deux sous-expressions. La première, **not(G(ALTITUDE > 30000))**, signifie que les avions à extraire n'ont pas une altitude constamment supérieure à 30000 pieds. A noter que cette sous expression aurait alors pu s'être présentée sous la forme alternative **F(ALTITUDE > 34000)**. La seconde sous expression, **and F(ALTITUDE > 34000 and LONGITUDE < 2.30)**, signifie que les avions à extraire ont volé au moins une fois à une altitude supérieure à 34000 pieds et ont eu au moins une fois une longitude inférieure à 2.30 (vis-à-vis de la France, des avions qui ont volé au sud de Paris). Il n'est cependant pas encore possible d'exprimer selon si les deux conditions de la seconde sous-expression (altitude et longitude) doivent être vérifiées en même temps, c'est-à-dire *les avions ayant volé au moins une fois au sud de Paris à plus de 34000 pieds*, ou indépendamment l'une de l'autre, c'est-à-dire *les avions qui ont volé au moins une fois au sud de Paris, et au moins une fois à 34000 pieds*. Ainsi, les expressions **F(ALTITUDE > 34000 and LONGITUDE < 2.30)** et **F(ALTITUDE > 34000) and F(LONGITUDE < 2.30)** sont actuellement évaluées de manière similaire. Il est actuellement envisagé de faire évoluer le langage pour intégrer ces nuances et offrir un plus grand pouvoir d'expression.

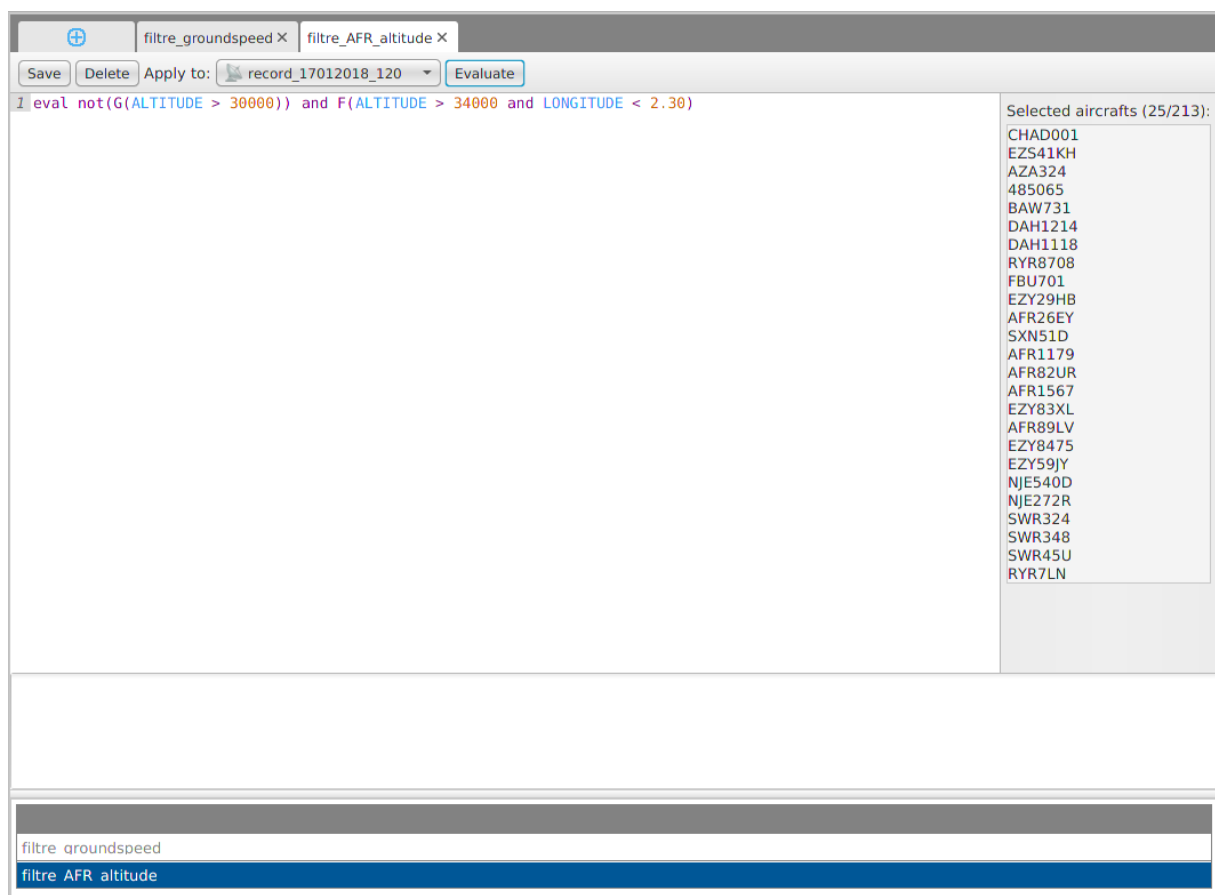


Figure 29 - Éditeur de filtres

Un filtre est générique, mais il doit cependant être appliqué à un enregistrement pour évaluer sa « portée ». Un menu déroulant dans la barre d'outils liste tous les enregistrements trouvés dans l'environnement de travail. Lorsqu'un enregistrement est sélectionné, le bouton d'évaluation devient cliquable et il est alors possible d'observer la portée du filtre sur l'enregistrement choisi. Le filtre exprimé en Figure 29 et décrit plus haut permet l'extraction de 25 avions sur les 213 présents sur l'enregistrement sélectionné. L'évaluation de filtre est très rapide, inférieure à 5 secondes pour les enregistrements les plus conséquents (> 2 heures).

Enfin, un système d'onglets (comme visible en haut de la Figure 29) et une mini bibliothèque de filtre (en bas de la figure), apporte une aide supplémentaire à la conception de filtres.

4.2. Éditeur de déclencheurs d'altération

Les avions concernés par un scénario donné sont extraits au moyen de filtres. Pour poursuivre l'élaboration d'un scénario, l'étape suivante (et toutefois optionnelle) consiste à définir à quels moments les actions du scénario seront effectuées. Par exemple, il n'est peut-être pas nécessaire d'altérer les avions ciblés que lorsque ceux-ci se trouvent dans une zone donnée, ou lorsque leur vitesse est inférieure à une certaine valeur. Ou encore, il se peut que le scénario

consiste à effectuer certaines altérations dès lors que tous les avions se trouvent au-dessus d'une certaine altitude, mais pas avant.

Pour assurer ces différents aspects, un langage pour la définition de déclencheurs d'altération a été créé, ainsi qu'un éditeur pour assister à leur conception. Le langage utilisé est très similaire au langage de création de filtres : il s'agit d'exprimer des conditions booléennes vis-à-vis des propriétés des avions. Ceci-dit, quelques variations différencient les deux langages, à commencer par l'absence d'opérateurs de logique temporelle LTL (G et F) dans le langage de déclencheurs. En revanche ce dernier possède deux types d'opérateurs, absents du langage de filtres, pour permettre d'exprimer les conditions de déclenchement.

D'une part, des opérateurs de déclenchements sont introduits dans le langage pour définir à quelle action (altérer, ne pas altérer) est associée l'évaluation positive d'une expression booléenne. Afin d'apporter une certaine souplesse dans la conception de déclencheurs, quatre opérateurs de déclenchement sont introduits :

- When (expr) : les altérations du scénario ne sont effectuées **que lorsque** *expr* est vraie.
- Not_when (expr) : les altérations du scénario ne sont effectuées **que lorsque** *expr* est fausse.
- As_soon_as (expr) : les altérations du scénario sont effectuées **dès lors que** *expr* est vraie. Ainsi, même si *expr* est fausse après avoir été évaluée à vrai en premier lieu, les altérations seront effectuées.
- Until (expr) : les altérations du scénario sont effectuées **jusqu'à ce que** *expr* soit vraie. Ainsi, même si *expr* est vraie après avoir été évaluée à faux en premier lieu, les altérations ne seront pas effectuées.

Toute expression booléenne doit être dans la portée d'un opérateur de déclenchement. De plus, il est possible de combiner des opérateurs au moyen des opérateurs logiques de conjonction et disjonction.

D'autre part, le langage de déclencheur d'altération propose trois types de contexte pour la définition d'expressions. L'objectif est de pouvoir exprimer des conditions pas seulement centrées sur un seul avion, mais sur un groupe d'avions, voir toute la *situation aérienne générale*. Ces contextes sont exprimés sous forme de mots-clés placés en préfixe des propriétés impliquées dans une expression. Il existe actuellement trois types de contexte :

- RAP.PROPERTY : la propriété PROPERTY porte sur toute la situation aérienne générale. Cela signifie que, à un instant donné, la propriété PROPERTY (par exemple, la latitude) de tous les avions doit respecter la condition pour que celle-ci soit évaluée positivement.

- TARGET.PROPERTY : Ce contexte est similaire au précédent, à ceci près que ne sont inclus dans l'évaluation de la condition que les avions qui ont été précédemment filtré, c'est-à-dire les avions sur lesquels porte le scénario.
- AIRCRAFT.PROPERTY : chaque avion est évalué individuellement.

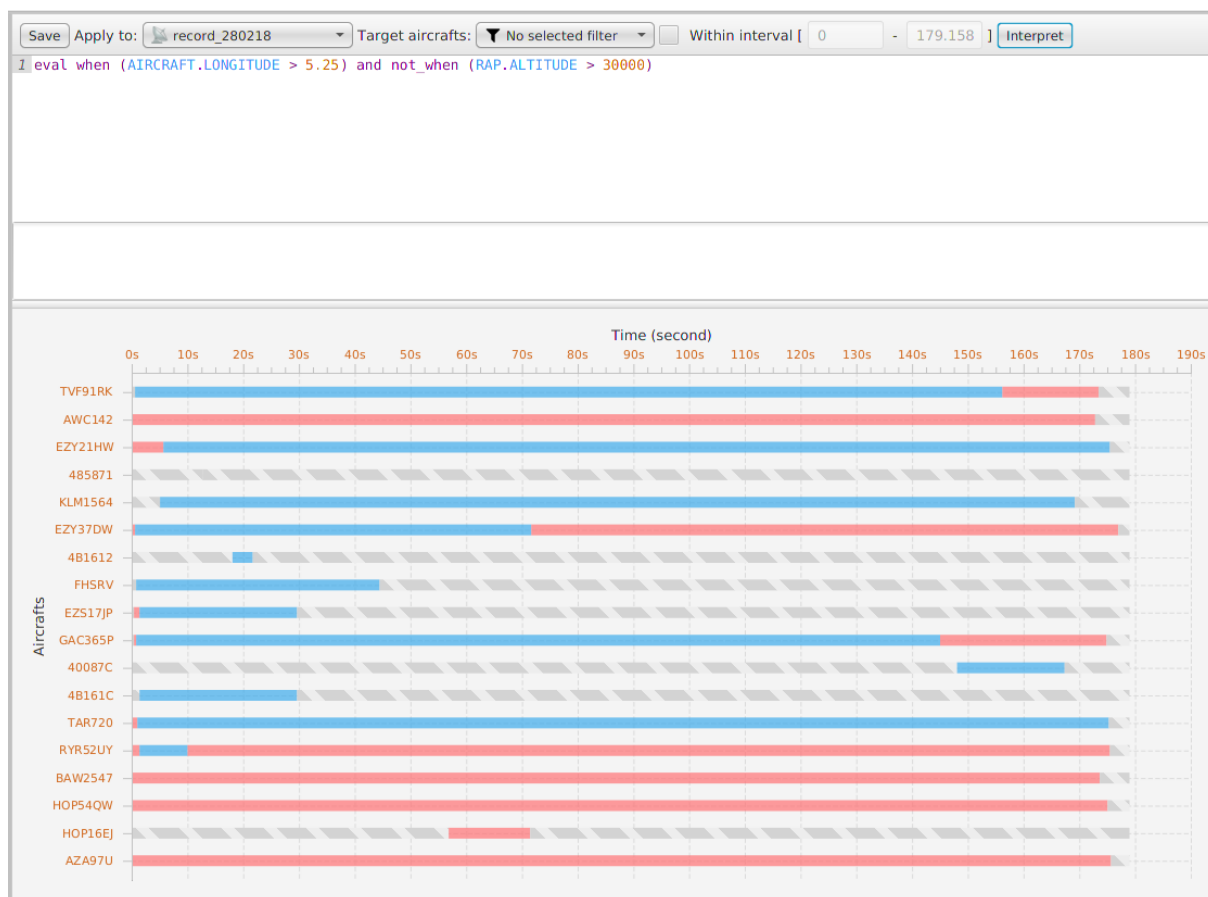


Figure 30 - Éditeur de déclencheurs d'altération

L'éditeur de filtre est présenté en Figure 30. La partie centrale permet l'écriture de conditions pour déclenchement d'altérations. Comme le langage de filtre, la conception est aidée par des mécanismes de coloration syntaxique, d'auto-complétions et d'aide à la correction des erreurs. Le déclencheur décrit en Figure 30 est comme suit :

eval when (AIRCRAFT.LONGITUDE > 5.25) and not_when (RAP.ALTITUDE > 30000)

On retrouve le mot clé **eval**, exprimant le début d'une expression de déclenchement. L'expression présentée est une conjonction de deux sous-expressions. La première, **when (AIRCRAFT.LONGITUDE > 5.25)**, signifie que les avions ne seront altérés (selon le scénario) que s'ils se trouvent à une longitude inférieure à 5.25. La seconde sous-expression, **not_when (RAP.ALTITUDE > 30000)**, signifie qu'en plus de la première condition, il ne faut pas que tous les avions aient une altitude supérieure à 30000 pieds. Ainsi, utiliser cette expression de déclenchement dans un scénario permet de n'effectuer les altérations sur les avions ciblés

que lorsque ceux-ci se trouvent à une longitude inférieure à 5.25 et qu'au moins un avion parmi la situation aérienne générale vole à moins de 30000 pieds.

Pour pouvoir évaluer un déclencheur, il faut l'appliquer à un enregistrement via le menu déroulant présent dans la barre d'outils. De plus, il est possible d'appliquer un filtre préalablement à l'évaluation du déclencheur, de manière à pouvoir observer le résultat uniquement sur les avions ciblés. Enfin, le bouton « evaluate » permet de calculer la portée du déclencheur sur les avions ciblés. Les résultats sont présentés sous la forme d'une ligne de temps, comme montré en bas de la Figure 30. Chaque avion possède sa propre ligne de temps, qui exprime à quels moments l'avion sera affecté par le scénario. La couleur **rouge** signifie qu'à cet instant de l'enregistrement, l'avion concerné ne sera pas affecté. En revanche, la couleur **bleue** signifie que l'avion sera affecté à cet instant précis. Enfin, les zébrures indiquent qu'à ces instants de l'enregistrement, l'avion ne fait pas partie de la situation aérienne générale.

4.3.Éditeur textuel de scénarios

En plus de la conception graphique, les scénarios peuvent être créés textuellement. Ceci permet de définir un modèle de scénario générique, car applicable à n'importe quel scénario, de plus, ce moyen de conception permet d'utiliser les filtres et les déclencheurs présentés dans les deux parties précédentes.

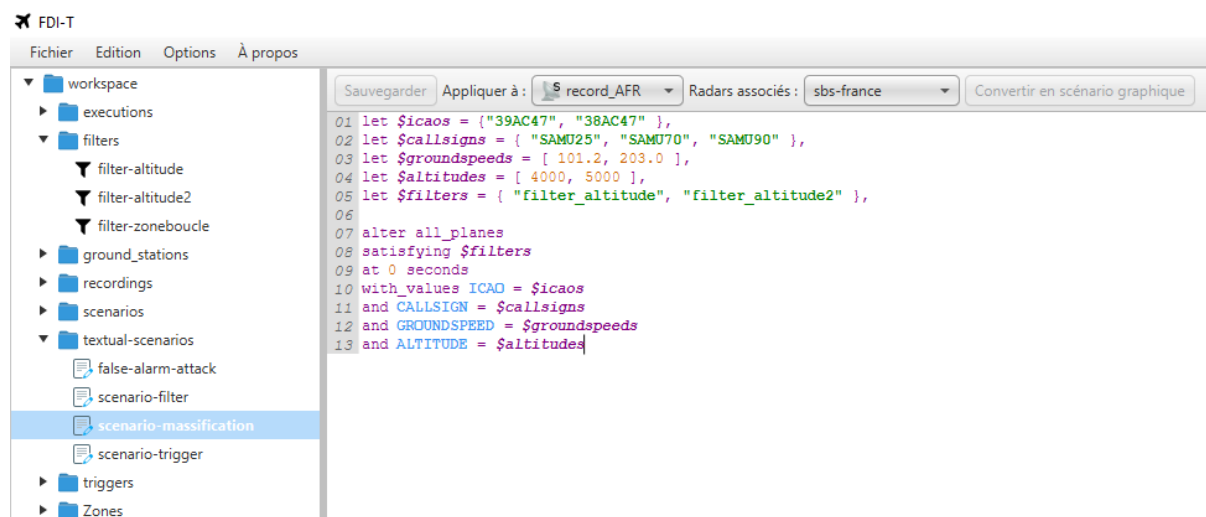


Figure 31 : Éditeur textuel de scénarios

L'éditeur textuel de scénario visible en Figure 31 offre deux fonctionnalités :

- La conversion d'un scénario textuel en scénario graphique.
- L'exécution d'un scénario textuel directement depuis l'éditeur d'exécution.

Dans la Figure 31, on peut voir que la barre d'outils de l'éditeur contient trois éléments :

- La liste déroulante contient la liste des enregistrements chargés dans l'espace de travail et permet de choisir celui à partir duquel la conversion en scénario graphique aura lieu.
- Le bouton « *Convertir en scénario* graphique » lance l'interprétation du langage dédié pour générer le scénario graphique correspondant. Ce bouton est cliquable seulement si le scénario respecte bien la grammaire décrite en Annexe 1 : Spécification du langage dédié et si un enregistrement a été sélectionné à partir de la liste déroulante.
- Le bouton « *Sauvegarder* » enregistre le scénario textuel.

5. ÉDITEUR DE ZONES

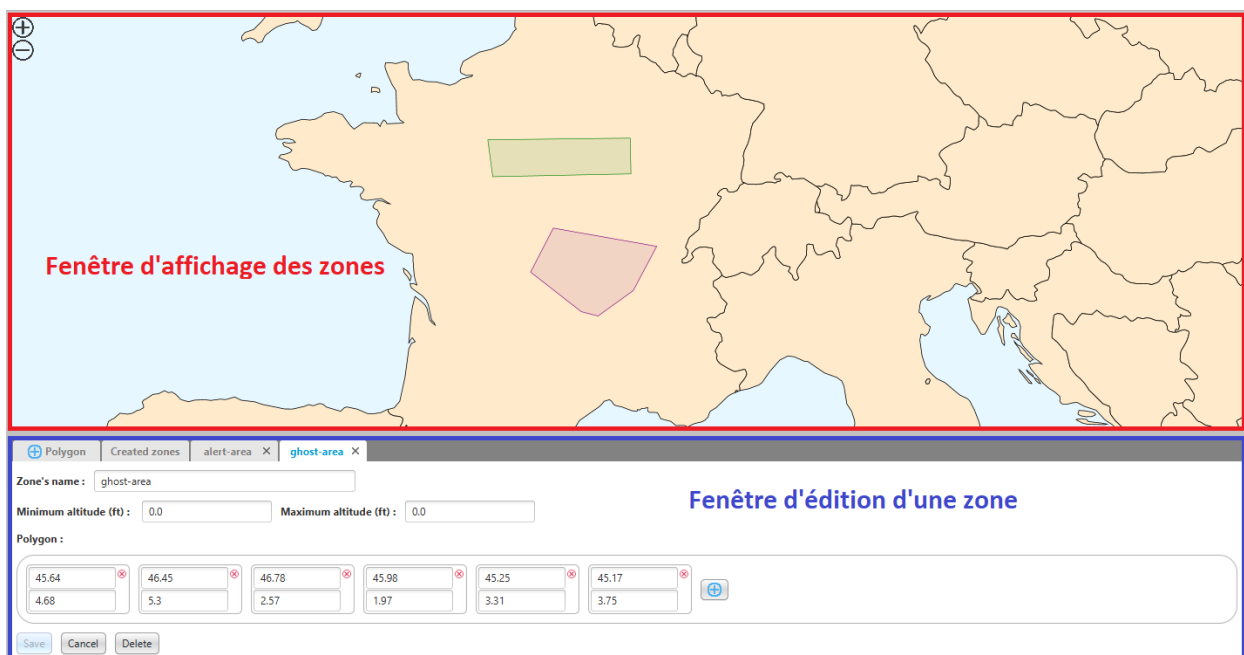


Figure 32 : Éditeur de zones

L'éditeur est divisé en deux parties :

- Une fenêtre pour afficher les zones en cours d'édition.
- Une fenêtre d'édition d'une zone.

Les deux fenêtres sont synchronisées, ainsi lors de la modification des coordonnées d'une zone, cette modification est aussi réalisée sur la carte.

Dans l'onglet d'édition d'une zone, il est possible de renommer la zone (**le nom ne doit pas être déjà utilisé par une zone dans le même dossier**), l'altitude minimum et maximum ainsi que les différentes coordonnées pour définir les frontières de la zone. Il existe deux types de zone :

- Les zones en polygone qui sont définies par un ensemble de points.

- Les zones en cercle qui sont définies par un centre et un rayon.

Nom de la zone : Nouvelle zone

Altitude minimum (ft) : 0 **Altitude maximum (ft) :** 0

Polygone : Bouton de suppression d'une coordonnée

47.74	46.67	45.59
2.58	0.35	3.52

Bouton d'ajout d'une coordonnée

Sauvegarder Annuler Supprimer

Figure 33 : Éditeur de zones – Onglet d'édition d'une zone en polygone

Le bouton « *Sauvegarder* » enregistre les changements et les réalise dans les différents éditeurs. Tant que les changements ne sont pas sauvegardés, ils ne sont pas effectifs dans les autres éditeurs.

Le bouton « *Annuler* » annule tous les changements effectués depuis la dernière sauvegarde.

Le bouton « *Supprimer* » supprime la zone.

Cercle :

Centre X: 47.59 Y: 4.16 Rayon: 50.0

Figure 34 : Éditeur de zones - Onglet d'édition d'une zone circulaire

La différence entre l'édition d'une zone en polygone et d'une zone circulaire concerne l'encadré dans lequel se trouve la liste des coordonnées en Figure 33. Pour une zone circulaire, il se présente comme celui de la Figure 34.

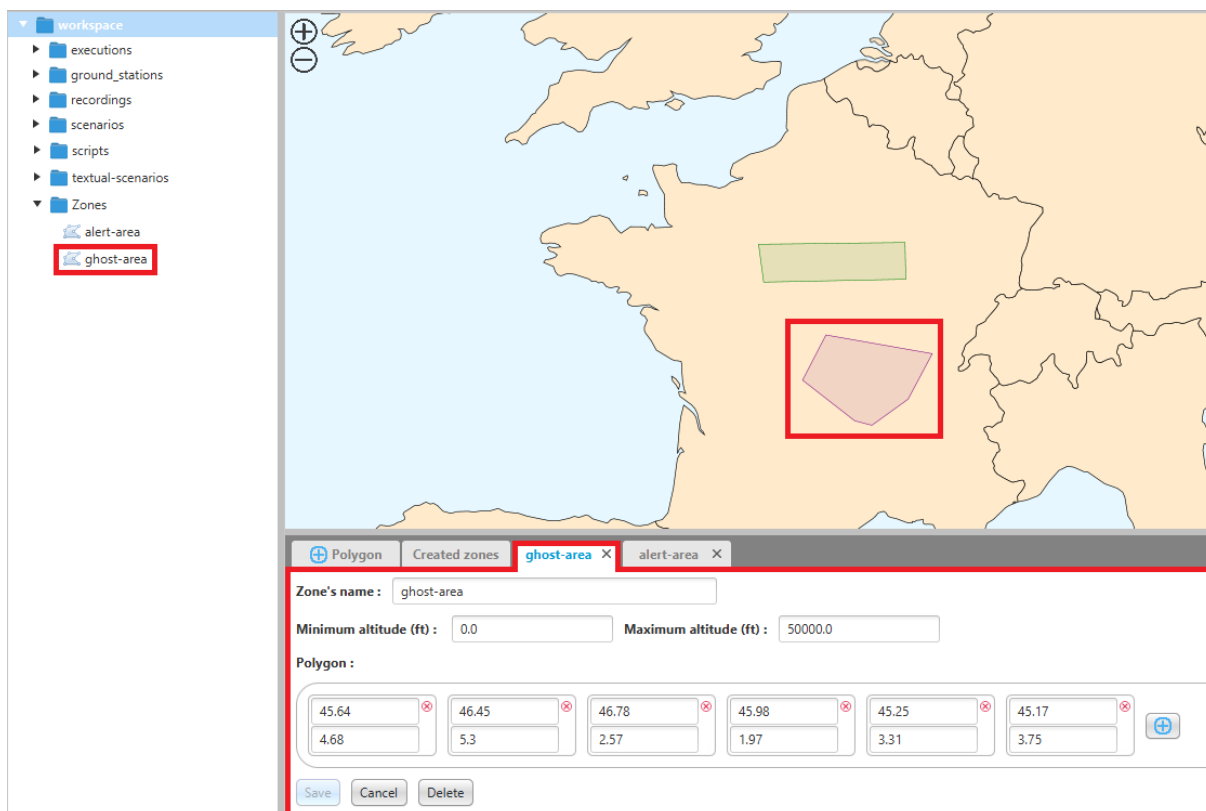


Figure 35 : Éditeur de zones – Exemple de zone créée

Il est également possible de voir dans l'onglet « Zones créées » toutes les zones existantes, avec les informations relatives à celles-ci. Cliquer sur le nom de la zone ouvre l'onglet d'édition de celle-ci.

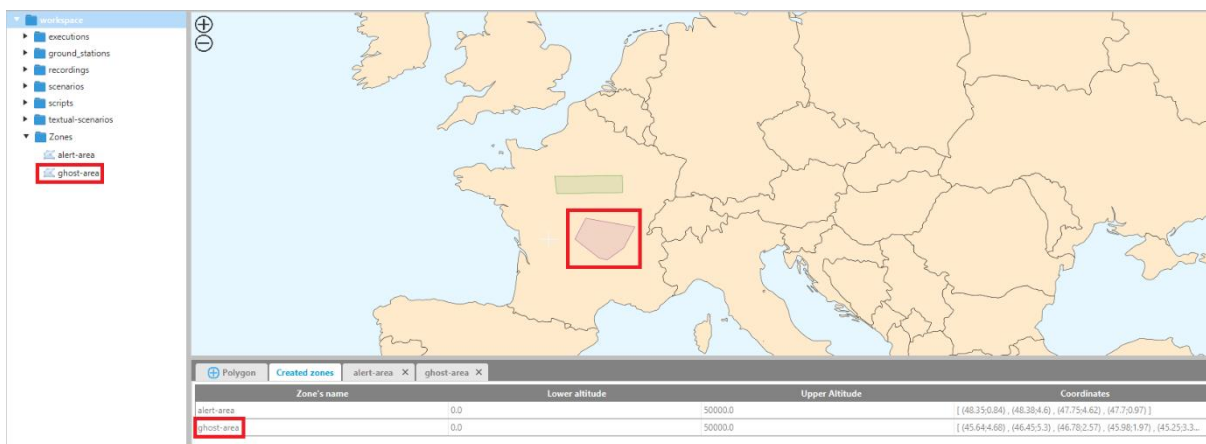


Figure 36 : Éditeur de zones – Tableau récapitulatif des zones créées

Les onglets avec l'icône « + » permettent de créer une nouvelle zone depuis l'éditeur de zone. Pour l'instant, tous les fichiers associés à une zone sont créés dans le dossier racine.

6. ÉDITEUR DE CONFIGURATIONS D'EXECUTIONS

6.1. Configuration serveur/client pour l'exécution de scénarios

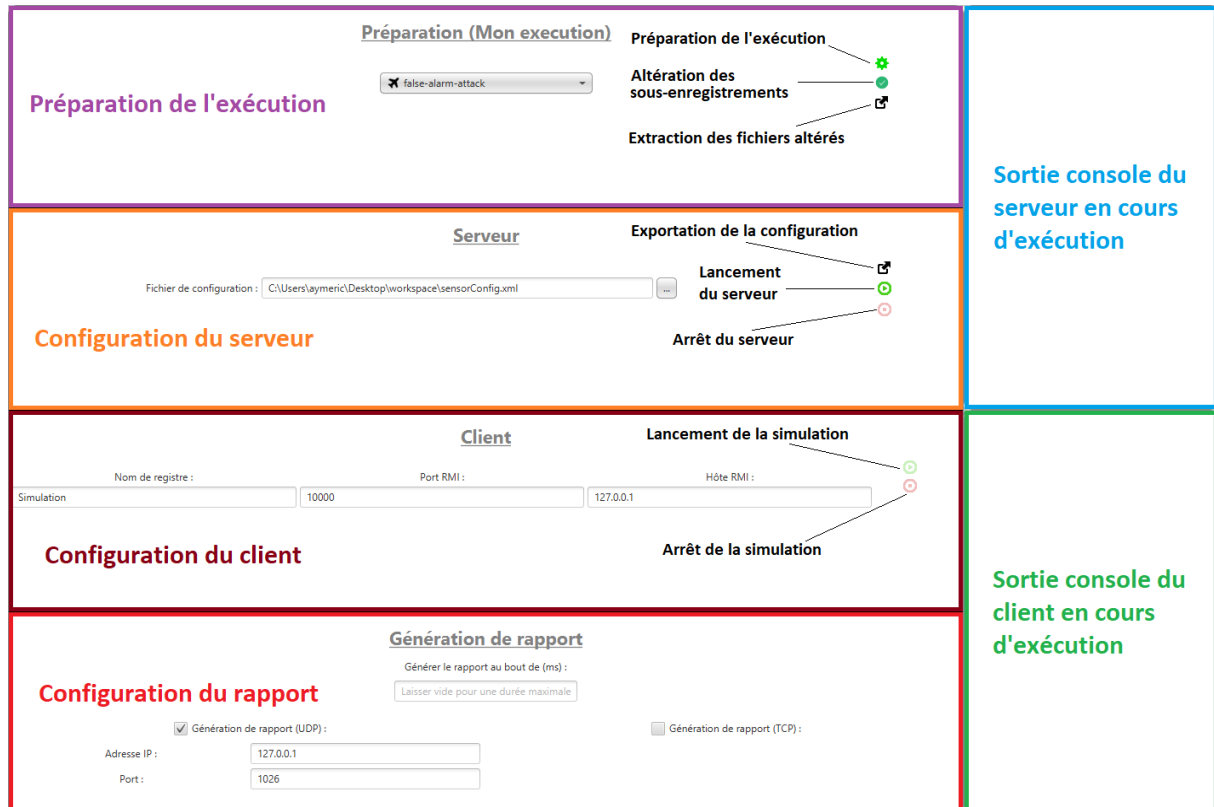


Figure 37 : Éditeur de configurations d'exécutions

L'éditeur d'exécution permet la concrétisation des scénarios, la **Erreur ! Source du renvoi introuvable.** montre que cet éditeur est divisé en quatre parties.

6.1.1. PARTIE « PREPARATION »

Cette partie, visible en Figure 38 permet à l'utilisateur de choisir le scénario à exécuter grâce à une liste déroulante et à découper l'enregistrement utilisé par le scénario choisi en fonction des radars associés à ce dernier.

Préparation (Mon exécution)



Figure 38 - Partie "Préparation" de l'éditeur d'exécution : exécution non préparée

Le serveur ne peut pas être lancé tant que l'exécution n'a pas été préparée, cette préparation se fait avec le bouton en roue dentée labelisé « *Préparation de l'exécution* » sur la Figure 37. Il

apparaît en noir sur la Figure 38, cela signifie que l'exécution n'a pas encore été préparée, quand l'utilisateur clique sur ce bouton, l'enregistrement est découpé en plusieurs fichiers (un par radar), le bouton devient vert et l'exécution passe en état « préparée » comme sur la Figure 39.

Préparation (Mon execution)



Figure 39 - Partie "Préparation" de l'éditeur d'exécution : exécution préparée

Une exécution reste dans l'état « préparée » tant que le scénario et les radars associés ne sont pas modifiés. Le cas échéant, le bouton de préparation redevient noir et l'exécution doit être préparée à nouveau pour pouvoir lancer le serveur.

Quand l'exécution est préparée, il est possible d'utiliser le bouton labélisé « Altération des sous-enregistrements » sur la Figure 37. Ce bouton permet d'appliquer les altérations sur les fichiers issus de la préparation de l'exécution, quand les altérations ont été appliquées, le bouton apparaît en vert, tel que sur la Figure 40, et l'exécution est dans l'état « altérée ».

Préparation (Mon execution)



Figure 40 - Partie "Préparation" de l'éditeur d'exécution : enregistrements altérés

À partir de l'état « altérée » il est possible de revenir à l'état « non altérée » en cliquant à nouveau sur le bouton d'altération. Cet état définit si les altérations définies dans le scénario seront appliquées aux enregistrements ou non lors du jeu.

Le bouton labélisé « Extraction des fichiers altérés » dans la Figure 37 permet de récupérer les sous-enregistrements avec les altérations appliquées. Pour pouvoir utiliser cette fonctionnalité, il faut que l'exécution soit dans les états « préparée » et « altérée ».

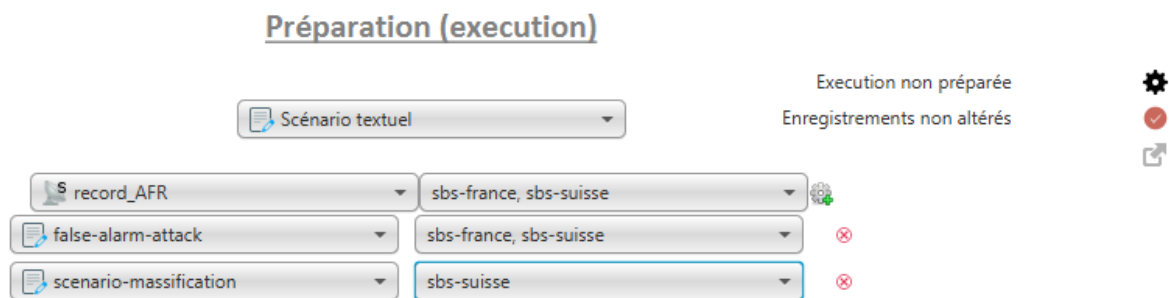


Figure 41 - Partie "Préparation" de l'éditeur d'exécution : le scénario textuel

La Figure 41 montre l'exemple d'un cas d'exécution de scénario textuel, en choisissant l'option « Scénario textuel » dans la liste déroulante, l'utilisateur pourra définir un scénario « à la volée » et choisir :

- L'enregistrement sur lequel joué le scénario, dans l'exemple ci-dessus « record_AFR ».
- Les radars associés à ce scénario, « sbs-france » et « sbs-suisse ».
- La liste des altérations à appliquer à l'enregistrement, « false-alarm-attack » et « scenario-massification », à noter que ces actions sont créées avec le DSL depuis l'Éditeur textuel de scénarios.
- Les radars ciblés par chaque altération, dans l'exemple on voit que les radars « sbs-france » et « sbs-suisse » sont ciblés par l'altération « false-alarm-attack » tandis que seul le radar « sbs-suisse » est ciblé par l'altération « scenario-massification ».

6.1.2. LA PARTIE « SERVEUR »

Cette partie, visible en Figure 42 permet d'exporter la configuration actuelle ou de lancer le serveur si l'exécution est dans l'état « préparée ».

Le bouton labelisé « *Lancement du serveur* » apparaît grisé sur la **Erreur ! Source du renvoi introuvable..** Cela signifie que l'exécution n'a pas été préparée ou que le fichier de configuration n'est pas bien formé. Pour les mêmes raisons, le bouton labelisé « *Exportation de configuration* » apparaît également grisé sur la **Erreur ! Source du renvoi introuvable..**



Figure 42 - Partie "Serveur" de l'éditeur d'exécution

Sur la Figure 42, les deux boutons sont cliquables, cela signifie que l'exécution est dans l'état « préparée » et que le fichier de configuration est bien formé.

L'exportation de la configuration permet à l'utilisateur de créer une archive au format ZIP contenant tous les fichiers nécessaires à reproduire l'exécution sans passer par l'interface graphique.

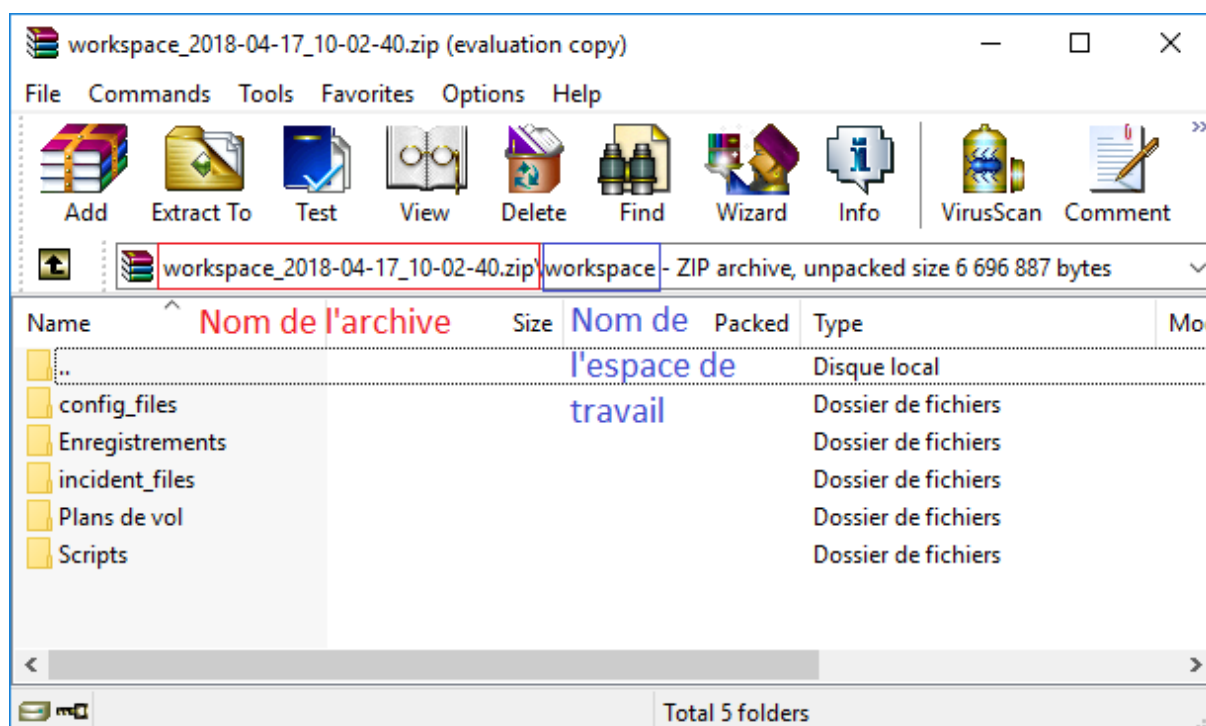


Figure 43 - Archive créée à la suite de l'exportation ouverte avec le logiciel WinRAR

Quand l'utilisateur clique sur le bouton d'exportation, il est invité à choisir une destination. À cette destination, une archive (voir exemple de la Figure 43) est créée. Elle contient les fichiers suivants :

- Dans le dossier « *config_files* » se trouvent les fichiers de configuration utilisés en **Erreur ! Source du renvoi introuvable.**
- Dans le dossier « *Enregistrements* » se trouve le fichier SBS ou BST sur lequel est joué le scénario sélectionné pour la concrétisation.
- Dans le dossier « *incidents_files* » se trouvent les fichiers incidents contenant les instructions sur les actions à réaliser sur l'enregistrement.
- Dans le dossier « *Plans de vol* » se trouve, s'il existe, le plan de vol lié à l'enregistrement.
- Dans le dossier « *Scripts* » se trouvent les fichiers de script Groovy qui ont été utilisés pour définir des assertions globales sur le scénario sélectionné pour la concrétisation.

6.1.3. LA PARTIE « CLIENT »

Cette partie, visible en Figure 44 permet à l'utilisateur de lancer l'exécution du test. Pour cela il devra spécifier les trois informations suivantes dans les champs de texte correspondant :

- Nom de registre RMI
- Numéro du port RMI
- Adresse de l'hôte RMI

Le bouton labelisé « *Bouton de lancement de la simulation* » dans la **Erreur ! Source du renvoi introuvable.** apparaît grisé car le serveur n'a pas encore été lancé. Le test ne peut être démarré que si le serveur a été préalablement lancé.

Client

Nom de registre :	Port RMI :	Hôte RMI :	▶
Simulation	10000	127.0.0.1	▶

Figure 44 - Partie "Client" de l'éditeur d'exécution

Dans la Figure 44, le bouton de lancement du client n'est plus grisé car le serveur est démarré. L'utilisateur peut alors cliquer sur le bouton vert pour commencer l'exécution du test, faisant ceci, il concrétise le scénario créé et l'envoi au serveur, le schéma de la Figure 45 décrit le fonctionnement de cette concrétisation.

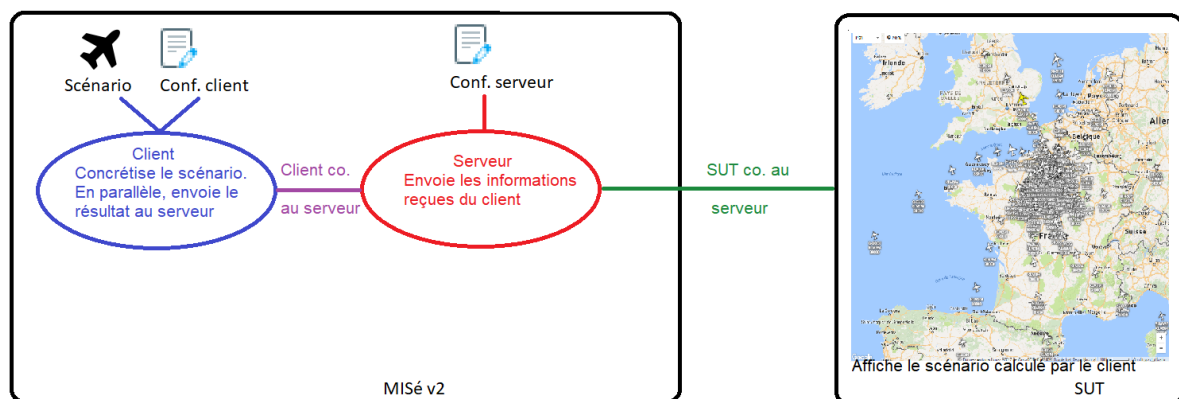


Figure 45 : Éditeur d'exécution - Fonctionnement de la concrétisation

Pour lancer l'exécution d'une attaque, il faut d'abord lancer le serveur et attendre que l'appareil à tester et le serveur soient connectés. Les messages suivants doivent apparaître dans la sortie console du serveur :

```
16:40:25 : ---- SERVER Started ----
16:40:29 : Someone's connected : [id:
0x4f23764e, L:/127.0.0.1:31003 -
R:/127.0.0.1:63244]
16:40:29 : Adding channel : [id:
0x4f23764e, L:/127.0.0.1:31003 -
R:/127.0.0.1:63244] - 4f23764e
```

Figure 46 : Éditeur d'exécution – Sortie console du serveur

Ensuite, il est possible de lancer l'exécution. La sortie de la console du client doit être la suivante si tout se déroule correctement :

```
16:45:15 : ### Threads Model & Replay
started... ###

00:09
```

Figure 47 : Éditeur d'exécution – Sortie console du client

Pour arrêter le serveur ou le client, il suffit de cliquer à nouveau sur les boutons d'arrêts correspondants, sur la **Erreur ! Source du renvoi introuvable.**, ils sont labelisés « *Bouton d'arrêt du serveur* » et « *Bouton d'arrêt du client* ».

6.1.1. LA PARTIE « GENERATION DE RAPPORT »

Cette partie, visible en Figure 48 permet à l'utilisateur d'indiquer si un rapport consignait les résultats de test devra être généré à l'issue de la concrétisation.

Génération de rapport

<input type="checkbox"/> Génération de rapport UDP : Adresse IP : <input type="text"/> Port : <input type="text"/>	<input checked="" type="checkbox"/> Génération de rapport TCP : Adresse IP : <input type="text"/> Port : <input type="text"/>
--	---

Figure 48 - Partie "Génération de rapport" de l'éditeur d'exécution

Dans cette partie l'utilisateur peut choisir de générer le rapport à partir d'informations reçues via la boucle de retour en utilisant le protocole TCP, UDP, ou en utilisant les deux simultanément. Pour chacun des protocoles, il devra spécifier un port et une adresse IP sur lesquelles sera envoyé le rapport généré.

6.2. Configuration d'un sensor de type SDPS

Il est possible de faire de la simulation de pistes système au format ASTERIX CAT062. Ainsi, en configurant un sensor de type SDPS dans le fichier « sensorConfig.xml », il est possible d'injecter des données pistes au format ASTERIX CAT062-065 au système à tester.

Pour ce faire, il suffit de modifier la configuration du fichier sensorConfig.xml comme suit :

```

<sensor type="SDPS">
  <sID>6</sID>
  <parameters>
    <ip>224.1.1.2</ip>
    <port>2416</port>
    <longitude>2.37</longitude>
    <latitude>48.85</latitude>
    <altitude>0</altitude>
    <sic>104</sic>
    <sac>8</sac>
    <callsign>true</callsign>
  </parameters>
</sensor>

```

Comme pour les autres types de sensor, celui-ci est de type SDPS, il faut renseigner les champs suivants :

- sID : ID du sensor (qui doit être différent des autres sensors définis dans le fichier)
- L'adresse IP (UDP multicast)
- Numéro de port (le système sous test devra écouter sur ce port et sur l'IP multicast renseignée)
- Longitude/latitude/altitude : position géographique du système de surveillance des pistes du système sous test qui reçoit les données au format ASTERIX CAT062-065
- SAC/SIC : identification du SDPS
- Callsign : Ce champ permet de définir si le callsign doit être transmis dans les messages ASTERIX CAT062 (item I062/390 champ Callsign)

Lors du lancement d'une exécution, on devrait alors voir apparaître des trames ASTERIX dans Wireshark par exemple :

787 8.905784	192.168.10.117	224.1.1.2	ASTERIX
794 8.931622	192.168.10.117	224.1.1.2	ASTERIX


```

> Frame 687: 85 bytes on wire (680 bits), 85 bytes captured (680 bits) on interface 0
> Ethernet II, Src: HewlettP_d7:73:4c (c8:d3:ff:d7:73:4c), Dst: IPv4mcast_01:01:02 (01:00:5e:01:01:02)
> Internet Protocol Version 4, Src: 192.168.10.117, Dst: 224.1.1.2
> User Datagram Protocol, Src Port: 53705, Dst Port: 2416
▼ ASTERIX packet, Category 062
  Category: 62
  Length: 43
  > Asterix message, #01, length: 40

```

Figure 49 : Trames ASTERIX CAT062

6.3. Configuration d'un sensor modélisant un radar

Il est possible de simuler des radars de type :

- Beast,
- SBS,

- PSR (primaire),
- SSR(secondaire),
- Mode S.

Figure 50 : Définition de radars

6.4. Configuration du serveur de logs **[NON FONCTIONNEL en V4.0.2]**

Il est possible d'activer un serveur de logs, permettant de recevoir en retour du système sous test, des logs au format SYSLOG, ces logs permettent d'être comparés avec les assertions, pour valider la détection d'anomalies injectées sur le système sous test à travers le scénario de test exécuté.

Pour configurer le serveur de logs, il faut modifier le fichier sensorConfig.xml de la façon suivante :

```
<logserverconf>
  <ip>xxx.xxx.xxx.xxx</ip>
  <port>40000</port>
</logserverconf>
```

Ce nœud xml doit être ajouté après le nœud <rmiconf>.

7. LANCEMENT CONVERTISSEUR SANS GUI

Avant de pouvoir lancer une simulation, il est nécessaire de convertir les fichiers beast ou sbs.

7.1. Conversion d'un fichier BEAST

7.1.1. CONVERSION D'UN FICHIER BEAST VERS UN FICHIER JSON

Pour effectuer la conversion d'un fichier BEAST vers un fichier JSON, il suffit d'effectuer la commande donnée en exemple ci-dessous :

Exemple :

```
java -jar .\TLAS-converter-1.9.0-SNAPSHOT.jar -b input.beast -btjo output.json
```

7.1.2. CONVERSION D'UN FICHIER JSON VERS UN FICHIER BEAST

Pour effectuer la conversion d'un fichier JSON vers un fichier BEAST, il suffit d'effectuer la commande donnée en exemple ci-dessous :

Exemple :

```
java -jar .\TLAS-converter-1.9.0-SNAPSHOT.jar -eb input.json -ebo output.beast
```

7.1.3. CONVERSION D'UN FICHIER JSON VERS DES FICHIERS NJSON

Pour effectuer la conversion d'un fichier JSON vers des fichiers NJSON, il suffit d'effectuer la commande donnée en exemple ci-dessous.

Le fichier JSON en entrée provient d'une conversion d'un fichier BEAST. Et le fichier de configuration contient les données concernant les radars.

Les fichiers NJSON en sortie seront de la forme `beast_x.json`

Exemple :

```
java -jar .\TLAS-converter-1.9.0-SNAPSHOT-API-KER.jar -bjtn .\input.json -p .\config.xml
```

7.2. Conversion d'un fichier SBS

7.2.1. CONVERSION D'UN FICHIER SBS VERS UN FICHIER JSON

Pour effectuer la conversion d'un fichier SBS vers un fichier JSON, il suffit d'effectuer la commande donnée en exemple ci-dessous :

Exemple:

```
java -jar .\TLAS-converter-1.9.0-SNAPSHOT-API-KER.jar -s .\input.sbs -stjo output.json
```


7.2.1. CONVERSION D'UN FICHIER JSON VERS UN FICHIER SBS

Pour effectuer la conversion d'un fichier JSON vers un fichier SBS, il suffit d'effectuer la commande donnée en exemple ci-dessous :

Exemple :

```
java -jar .\TLAS-converter-1.9.0-SNAPSHOT-API-KER.jar -es input.json -eso output.sbs
```

7.2.1. CONVERSION D'UN FICHIER JSON VERS DES FICHIERS NJSON

Pour effectuer la conversion d'un fichier JSON vers des fichiers NJSON, il suffit d'effectuer la commande donnée en exemple ci-dessous.

Le fichier JSON en entrée provient d'une conversion d'un fichier SBS. Et le fichier de configuration contient les données concernant les radars.

Les fichiers NJSON en sortie seront de la forme sbs_x.json

Exemple :

```
java -jar .\TLAS-converter-1.9.0-SNAPSHOT-API-KER.jar -stn .\input.sbs -p .\config.xml
```

8. LANCEMENT SIMULATEUR SANS GUI

Il est possible d'utiliser le simulateur pour lancer des scénarios à partir des fichiers de configuration XML sans interface graphique. Cela nécessite de lancer le jar de l'application (en ligne de commande) dans 2 terminaux différents, un pour la partie « serveur » et un pour la partie « client », la partie serveur doit bien entendu être lancée avant la partie client.

8.1. Lancement du serveur

Pour lancer la partie serveur, il faut exécuter la commande suivante :

```
java -jar TLAS-simulator-1.0.1-SNAPSHOT.jar -s -cfg sensorConfig.xml
```

- sensorConfig.xml étant le fichier de configuration de serveur, avec la liste des radars définis.

8.2. Lancement du client

Pour lancer la partie client, il faut exécuter la commande suivante :

```
java -jar TLAS-simulator-1.0.1-SNAPSHOT.jar -c -rR <rmiRegistryName> -rH "<rmiServerIP:port>" (rmiServerIP and port are defined in sensorConfig.xml)
```

- rmiRegistryName : Le nom du register RMI défini dans sensorConfig.xml
- rmiServerIP et port sont également définis dans sensorConfig.xml

8.2.1. GENERATION DU RAPPORT DE TEST

Il est possible de générer un rapport de test sur les retours JSON de la boucle de retour du système sous test.

8.2.1.1. GENERATION DU CSV

Pour pouvoir générer les résultats du test il va falloir « activer » la génération des résultats dans le simulateur, en le configurant de la façon suivante :

- Fichier sensorConfig.xml :
 - Rajouter le nœud xml suivant après le nœud <rmiconf>

```
<feedbackserverconf>
  <ip>xxx.xxx.xxx.xxx</ip>→ ip de la machine
  <port>41000</port>→ port par défaut 41000 (port qui est celui du système sous test sur lequel il émet les retours JSON)
</feedbackserverconf>
```

8.2.1.2. GENERATION DU RAPPORT

Pour générer le rapport au format html d'un scénario de test, il suffit d'exécuter la commande suivante :

```
java -jar misereporting-1.1-RELEASE.jar -p "<pathToTestFolder>"
```

- pathToTestFolder : chemin du dossier contenant tous les fichiers .csv générés en sortie du simulateur FDI-T à la fin de l'exécution d'un scénario (si la génération de rapport a été activée pour ce scénario).

Le résultat de la commande est sauvegardé dans un dossier au format *report_aaaa-MM-dd*, aaaa-mm-dd étant la date du jour de l'exécution du test.

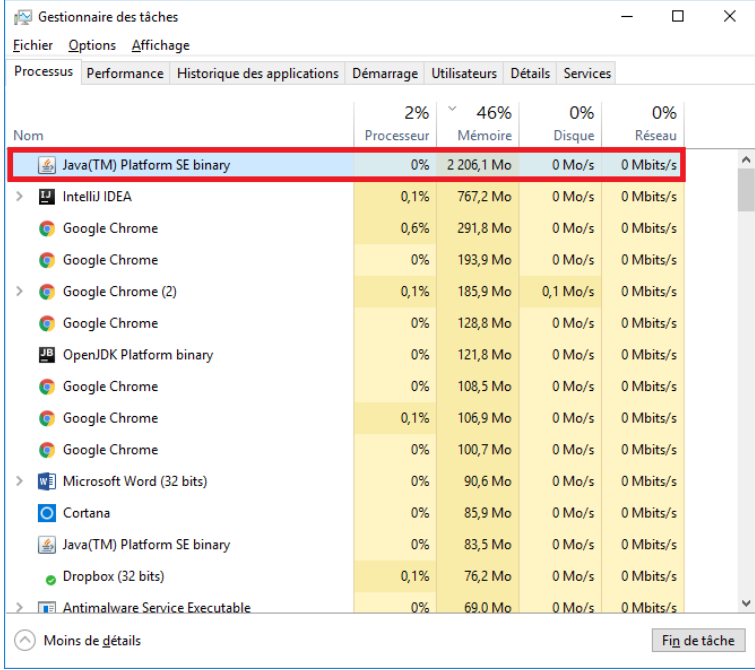
Dans ce dossier, plusieurs fichiers sont générés :

- report_aaaa-mm-dd.html : fichier principal du rapport de test
- Une liste de fichiers de la forme *assertion_i_aaaa-mm-dd.html* : ce sont les fichiers détaillés par assertion, i étant le numéro de l'assertion (dans l'ordre déclaré dans le scénario graphique sur FDI-T)

9. PROBLEMES CONNUS ET SOLUTIONS

Le client/serveur sont toujours en exécution alors que l'application a été arrêtée ou l'arrêt a été demandé.

Ouvrir le gestionnaire des tâches (Ctrl+Alt+Supp), et terminer le processus Java consommant le plus de mémoire.

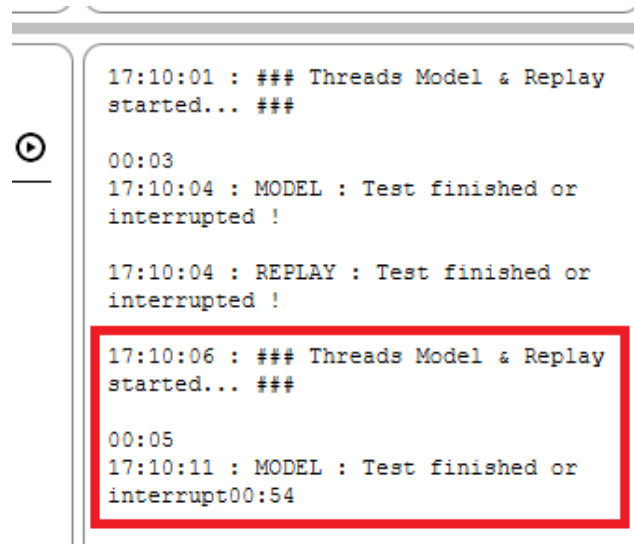


Nom	Processeur	Mémoire	Disque	Réseau
Java(TM) Platform SE binary	0%	2 206,1 Mo	0 Mo/s	0 Mbits/s
IntelliJ IDEA	0,1%	767,2 Mo	0 Mo/s	0 Mbits/s
Google Chrome	0,6%	291,8 Mo	0 Mo/s	0 Mbits/s
Google Chrome	0%	193,9 Mo	0 Mo/s	0 Mbits/s
Google Chrome (2)	0,1%	185,9 Mo	0,1 Mo/s	0 Mbits/s
Google Chrome	0%	128,8 Mo	0 Mo/s	0 Mbits/s
OpenJDK Platform binary	0%	121,8 Mo	0 Mo/s	0 Mbits/s
Google Chrome	0%	108,5 Mo	0 Mo/s	0 Mbits/s
Google Chrome	0,1%	106,9 Mo	0 Mo/s	0 Mbits/s
Google Chrome	0%	100,7 Mo	0 Mo/s	0 Mbits/s
Microsoft Word (32 bits)	0%	90,6 Mo	0 Mo/s	0 Mbits/s
Cortana	0%	85,9 Mo	0 Mo/s	0 Mbits/s
Java(TM) Platform SE binary	0%	83,5 Mo	0 Mo/s	0 Mbits/s
Dropbox (32 bits)	0,1%	76,2 Mo	0 Mo/s	0 Mbits/s
Antimalware Service Executable	0%	69,0 Mo	0 Mo/s	0 Mbits/s

Figure 51 : Erreur – L'application tourne toujours en tâche de fond

Le client/serveur sont toujours en exécution alors que l'arrêt a été demandé depuis l'éditeur.

Il arrive parfois que le client ne soit pas arrêté et que le chronomètre continue à tourner.



```
17:10:01 : ### Threads Model & Replay
started... ###

00:03
17:10:04 : MODEL : Test finished or
interrupted !

17:10:04 : REPLAY : Test finished or
interrupted !

17:10:06 : ### Threads Model & Replay
started... ###

00:05
17:10:11 : MODEL : Test finished or
interrupt00:54
```

Figure 52 : Erreur : le client fonctionne toujours

La seule solution est d'arrêter manuellement l'application comme pour le problème précédent.

La création de scripts groovy n'est pas possible dans la version actuelle du simulateur.

Il faut au préalable déposer des scripts Groovy (fichiers .gy) dans le dossier « Scripts » de FDI-T, même vierge et ensuite les éditer depuis l'interface.

Lors de l'ajout de nouveaux fichiers dans un workspace courant (ouvert dans le simulateur), les dossiers ne sont pas rafraichis avec les nouveaux fichiers.

Pour ce faire, il faut ouvrir le workspace à nouveau pour voir apparaitre les fichiers ajoutés.

10. TABLE DES FIGURES

Figure 1 : Interface graphique de FDI-T.....	5
Figure 2 : Exemple de structure d'un projet	7
Figure 3 : Système de fichiers - Fenêtre d'ouverture d'un projet.....	8
Figure 4 : Système de fichiers - Affichage d'un projet.....	8
Figure 5 : Éditeur de zones – Fenêtre de création d'une zone	9
Figure 6 : Système de fichiers - Affichage d'une zone	9
Figure 7 : Fenêtre de création d'un scénario	9
Figure 8 : Système de fichier - Affichage d'un scénario textuel.....	10
Figure 9 : Système de fichiers - Affichage d'un scénario	10
Figure 10 : Affichage d'un enregistrement et de ses scénarios rattachés	10
Figure 11: Éditeur de configurations d'exécutions - Fenêtre de création d'une configuration	11
Figure 12 : Système de fichiers -Affichage d'une configuration d'exécution.....	11
Figure 13 : Éditeur graphique de scénarios	12
Figure 14 : Éditeur graphique de scénarios - Fenêtre d'information	13
Figure 15 : Éditeur graphique de scénarios - Édition d'une action : paramètres généraux.....	14
Figure 16 : Éditeur graphique de scénario – Paramètres spécifiques d'une altération	15
Figure 17 : Éditeur graphique de scénarios - Exemple d'altération sur les attributs.....	15
Figure 18 : Éditeur graphique de scénarios - Tableau récapitulatif des altérations possibles..	16
Figure 19 : Massification par intervalle de valeurs	16
Figure 20 : Massification par liste de valeurs.....	16
Figure 21 : Éditeur graphique de scénarios – Paramètres spécifiques d'une saturation	17
Figure 22 - Éditeur graphique de scénario – Paramètres spécifiques d'une action personnalisée.....	18
Figure 23 : Éditeur graphique de scénarios – Onglet d'édition des filtres	19
Figure 24 : Éditeur graphique de scénarios - Exemples de filtres sur les attributs.....	20
Figure 25 - Exemple d'assertion globale	21
Figure 26 : Définition d'une assertion	21
Figure 27 - Éditeur graphique de scénarios – Affichage des plans de vol	22
Figure 28 - Éditeur graphique de scénario - Choix d'un fichier de plans de vol.....	22
Figure 29 - Éditeur de filtres	25
Figure 30 - Éditeur de déclencheurs d'altération	27
Figure 31 : Éditeur textuel de scénarios.....	28
Figure 32 : Éditeur de zones	29
Figure 33 : Éditeur de zones – Onglet d'édition d'une zone en polygone.....	30
Figure 34 : Éditeur de zones - Onglet d'édition d'une zone circulaire.....	30

Figure 35 : Éditeur de zones – Exemple de zone créée.....	31
Figure 36 : Éditeur de zones – Tableau récapitulatif des zones créées.....	31
Figure 37 : Éditeur de configurations d'exécutions.....	32
Figure 38 - Partie "Préparation" de l'éditeur d'exécution : exécution non préparée.....	32
Figure 39 - Partie "Préparation" de l'éditeur d'exécution : exécution préparée	33
Figure 40 - Partie "Préparation" de l'éditeur d'exécution : enregistrements altérés	33
Figure 41 - Partie "Préparation" de l'éditeur d'exécution : le scénario textuel	34
Figure 42 - Partie "Serveur" de l'éditeur d'exécution.....	34
Figure 43 - Archive créée à la suite de l'exportation ouverte avec le logiciel WinRAR	35
Figure 44 - Partie "Client" de l'éditeur d'exécution	36
Figure 45 : Éditeur de configurations d'exécutions - Fonctionnement de la concrétisation	36
Figure 46 : Éditeur de configurations d'exécutions – Sortie console du serveur	37
Figure 47 : Éditeur de configurations d'exécutions – Sortie console du client.....	37
Figure 48 - Partie "Génération de rapport" de l'éditeur d'exécution	37
Figure 49 : Trames ASTERIX CAT062.....	38
Figure 50 : Définition de radars	39
Figure 51 : Erreur – L'application tourne toujours en tâche de fond.....	43
Figure 52 : Erreur : le client fonctionne toujours.....	44

ANNEXE 1 : SPECIFICATION DU LANGAGE DEDIE

Le langage dédié, ou Domain Specific Language (DSL) permet de concevoir des scénarios textuellement.

Dans la version de FDI-T livrée avec cette documentation, la grammaire complète n'est pas encore interprétable par le logiciel et **l'utilisation d'éléments de grammaire hors de la portée de cette annexe mènera à une erreur.**

Le code couleur utilisé doit se comprendre de la manière suivante :

- Les éléments en **gras** représentent les **règles de production**.
- Les éléments **violet** représentent les **mots clés** du langage.
- Les éléments **BLEU MAJUSCULE** représentent les **ÉNUMÉRATIONS**.
- Les éléments **verts** représentent un **type attendu**.
- Les éléments **rouges** représentent les **connecteurs**. Leur signification est la suivante :
 - o **|** → OU exclusif
 - o ***** → L'élément apparait zéro ou plusieurs fois
 - o **?** → L'élément apparait zéro ou une fois
 - o **(et)** → Délimite la portée des connecteurs

instruction	→	alter target scope parameters (global assertion)? hide target scope (global assertion)? saturate target scope parameters (global assertion)?
target	→	plane criteria all_planes (criteria)?
criteria	→	criterion (and criterion)*
criterion	→	with binCriterion (INSIDE OUTSIDE) area satisfying String
binCriterion	→	aircraftCrit (= < > < > <= =>) (Float Integer)
aircraftCrit	→	MAX_ALTITUDE MIN_ALTITUDE CALLSIGN KNOWN_POSITIONS ICAO
area	→	area String
scope	→	at time (within String)? start end (within String)?
start	→	from time
end	→	until time
time	→	Integer seconds
parameters	→	with_values parameter (and parameter)*
parameter	→	aircraftParam = (String Integer Float) (assertion)?
aircraftParam	→	LONGITUDE LATITUDE EMERGENCY SPI ALTITUDE ICAO CALLSIGN SQUAWK GROUNDSPEED TRACK
assertion	→	assert String

Instruction :

alter target scope parameters(global assertion)? : définit des consignes pour altérer l'ensemble des caractéristiques définies dans la partie « **parameters** » du ou des appareil(s) spécifié(s) dans la partie « **target** », pendant une fenêtre de temps spécifiée dans « **scope** ».

saturate target scope parameters(global assertion)? : définit des consignes pour saturer la situation aérienne générale, l'ICAO des appareils générés est défini dans la partie « **parameters** ». Les avions cibles sont défini dans la partie « **target** », et la saturation aura lieu pendant une fenêtre de temps spécifiée dans « **scope** ».

hide target scope (global assertion)? : définit la suppression des appareils spécifiés dans la partie « **target** », pendant une fenêtre de temps spécifiée dans « **scope** ».

Target :

Il existe plusieurs façons de sélectionner des pistes ADS-B :

- **all_planes** : toutes les pistes ADS-B sont sélectionnées
- **plane criteria**: sélectionne une piste ADS-B respectant l'ensemble des critères listés dans « **criteria**»
- **all_planes (criteria)?** : sélectionne toutes les pistes ADS-B respectant l'ensemble des critères listés dans « **criteria**»

Criteria :

Les différents critères sont :

- **aircraftCrit(=|<>|<|>|<=|>=)(Float|Integer)** : la valeur d'un **aircraftCrit** doit être égale à celle demandée dans la partie « **Float** » ou « **Integer** ». L'élément **aircraftCrit** est une énumération, grâce à la complétion automatique de l'éditeur textuel de scénarios, l'utilisateur aura le choix entre tous les critères.
- **INSIDEarea** : l'appareil doit se trouver dans la zone géographique définie dans **area**.
- **OUTSIDE area** : l'appareil doit se trouver hors de la zone géographique définie dans **area**.
- **satisfying String** : l'appareil doit respecter les critères définis dans le filtre dont le nom est spécifié dans la partie « **String** ».

Area :

area String : sélectionne une zone existante qui porte le nom spécifié dans la partie « **String** »

Scope :

Les différentes fenêtres de temps sont :

- **at time (within String)?** : l'instruction démarre au temps « **time** » jusqu'à la fin de l'enregistrement. Si le mot clé « **within** » est utilisé, alors le nom d'un déclencheur doit être spécifié dans la partie « **String** ». L'instruction sera alors déclenchée à partir du temps « **time** » tout en respectant les conditions de déclenchement.
- **from time until time (within String)?** : l'instruction démarre à partir du temps « **time** » jusqu'à un horaire spécifié dans le deuxième « **time** ». Comme pour la fenêtre de temps précédente, il est possible de combiner cette instruction avec un déclencheur.

Assert :

Il existe deux façons de définir les assertions :

- `assert String` : spécifie dans « `String` » résultat attendu dans des retours syslog du système sous test, une assertion déclarée de la sorte est liée à une altération de **parameter**.
- `globalassert String` : l'effet est le même que la déclaration précédente à ceci près que l'assertion n'est pas liée à une altération mais au scénario entier.

Parameters :

`with_values parameter (and parameter)*` : est la liste de tous les paramètres.

`aircraftParam = (String|Integer|Float)(assertion)?` : le paramètre **aircraftParam** doit être altéré avec la valeur spécifiée dans « `String` », « `Integer` » ou « `Float` ». L'assertion spécifié dans « **assertion** » sera liée à ce paramètre.