

# Cas d'utilisations plugin IBM

## Cas 1 : Créer un nouveau projet et vpz, simuler la création d'un individu

1. Créer un nouveau projet
2. Créer un nouveau Vpz test\_0
3. Ouvrir test\_0.vpz avec le plugin IBM
4. Créer une nouvelle classe Loup avec un compartiment C1 = 1
  - (4.1) Cliquez-droit sur la colonne des classes, add.
  - (4.2) Entrer le nom "Loup"
  - (4.3) Le plugin Forrester s'ouvre, définir la dynamique de l'individu ici
  - (4.4) Valider
5. Ajouter au Script principal

```
ibm : addModel (1 , "Loup" )
```

6. Valider, le Controleur se crée
7. Mettre en place un observable avec une vue
  - (7.1) Créer une vue test\_0View, type file. Valider
  - (7.2) Ajouter la vue au compartiment C1 de Loup : Aller dans Atomic Model Loup, éditer obs\_DTE\_Loup
  - (7.3) Cocher le bouton radio, valider
  - (7.4) Enregistrer
8. Lancer la simulation / ! Il manque vle.extension.differential-equation dans les Build-Depends de Description.txt

## Cas 2 : Simuler la création de plusieurs individus

1. Ouvrir le test\_0.vpz avec le plugin IBM
2. Modifier la commande `ibm:addModel(1,"Loup")` pour créer 10 individus

```
ibm:addModel(10,"Loup")
```

3. Lancer la simulation

### Cas 3 : Création de plusieurs individus grâce aux conditions expérimentales

1. Ouvrir les conditions expérimentales
2. Ajouter un paramètre “nb\_Loup” avec pour valeur 5.
3. Ouvrir “test\_0.vpz” avec le plugin
4. Mettre dans le script

```
ibm:addModel("nb_Loup","Loup")
```

5. Simuler, remarquer 5 loups créés.

### Cas 4 : Création de plusieurs individus grâce aux conditions expérimentales

1. Ouvrir test\_0.vpz avec le plugin IBM
2. Mettre dans le script

```
ibm:addModelWithParam(3,"Loup", "C1", 15)
```

3. Simuler, remarquer le changement de valeur de C1

### Cas 5 : Plusieurs individus, modification d’un paramètre grâce aux conditions expérimentales

1. Ouvrir les conditions expérimentales
2. Ajouter le paramètre “dix” = 10 à cond.Controleur
3. Ouvrir test\_0.vpz avec le plugin IBM
4. Mettre dans le script

```
ibm:addModelWithParam(3,"Loup", "C1", "dix")
```

5. Simuler, remarquer le changement de valeur de C1

### Cas 6 : Plusieurs individus avec des paramètres distincts

1. Ouvrir le test\_0.vpz avec le plugin IBM
2. Mettre dans le script

```
for i=1,5,1 do
    ibm:addModelWithParam(1,"Loup", "C1", i)
end
```

3. Simuler, remarquer le changement de valeur de C1

## Cas 7 : Plusieurs individus, modification de plusieurs paramètres

1. Ouvrir test\_0.vpz avec le plugin IBM
2. Sélectionner Loup
3. Faire un clic droit et sélectionner Modify
4. Ajouter un compartiment C2 = 0
5. Mettre dans le script

```
for i=1,5,1 do
    ibm:addModelWithParam(1,"Loup", "C1", i, "C2", 3)
end
```

6. Ajouter un observable de C2
  - (6.1) Ouvrir la fenêtre de dialogue Atomic Model de Loup
  - (6.2) Aller dans l'onglet Observable et Editer obs\_DTE\_Loup
  - (6.3) Ajouter une vue à C2
7. Simuler, remarquer le changement de valeur de C1 et de C2

## Cas 8 : Créer un évènement d'initialisation

1. Ouvrir test\_0.vpz avec le plugin IBM
2. Ajouter un script grâce au bouton "+", l'appeler "init", valider
3. Une nouvelle ligne s'affiche avec le nom "init", cliquer sur la flèche, une zone de texte apparaît, c'est là que l'évènement "init" doit être décrit.
4. Couper-coller le texte du Script principal dans le script init.
5. Mettre dans le Script principal

```
ibm:addEvent("INIT", "init")
```

6. Simuler, remarquer le non changement depuis le cas précédent

## Cas 9 : Modifier les scripts depuis les conditions expérimentales

1. Ouvrir les conditions expérimentales
2. Dans cond\_Controleur remarquer les paramètres Script et init qui correspondent au script principal et à l'évènement "init" créés.
3. Modifier init par

```
for i=1,5,1 do
    ibm:addModelWithParam(1,"Loup", "C1", i)
end
```

Seuls les paramètres C1 seront modifiés

4. Valider
5. Simuler, remarquer que tous les C2 ont la valeur par défaut 0

## Cas 10 : Ajouter un évènement SINGLE

1. Ouvrir test\_0.vpz avec le plugin IBM
2. Ajouter un nouveau Script. L'appeler "single"
3. Dans la zone de script de "single" mettre

```
print("Il est : ", ibm:getTime());
```

4. Dans la zone de Script principal mettre

```
ibm:addEvent("SINGLE",50,"single");
```

5. Simuler, remarquer l'affichage de "Il est : 50" dans la console à la date 50.

## Cas 11 : Envoyer une valeur à un individu

1. Ouvrir le test\_0.vpz avec le plugin IBM
2. Dans le script de "single" créé précédemment ajouter

```
ibm:setModelValue("Loup", 3, "C1", 0)
```

3. Simuler, remarquer le changement de la valeur C1 du 3ème Loup à partir de la date 50.

## Cas 12 : Envoyer une valeur des conditions expérimentales à un modèle

1. Ouvrir le test\_0.vpz avec le plugin IBM
2. Dans le script de “single” créé précédemment ajouter

```
ibm:setModelValue("Loup", 3, "C1", "dix")
```

3. Simuler, remarquer le changement de la valeur C1 du 3ème Loup à partir de la date 50.

## Cas 13 : Envoyer une valeur à un individu grâce à son nom

1. Ouvrir le test\_0.vpz avec le plugin IBM
2. Dans le script de “single” créé précédemment ajouter

```
ibm:setModelValue("Loup_0", "C1", "dix")
```

3. Simuler, remarquer le changement de la valeur C1 de “Loup\_0” à partir de la date 50.

## Cas 14 : Obtenir la valeur d’un paramètre d’un individu et appel à plusieurs évènements SINGLE

1. Ouvrir le test\_0.vpz avec le plugin IBM
2. Créer un nouvel évènement SINGLE appelé “single2”
3. Dans le script de “single2” nouvellement créé, ajouter

```
print("valeur de C1:", ibm:getModelValue("Loup", 1,"C1"))
```

4. Modifier le Script principal par

```
ibm:addEvent("INIT","init")
ibm:addEvent("SINGLE",49,"single2");
ibm:addEvent("SINGLE",50,"single");
ibm:addEvent("SINGLE",51,"single2");
```

5. Simuler, remarquer la trace dans la console. /! Modifier une valeur et lui demander directement après sa valeur ne fonctionne pas.

### Cas 15 : Obtenir la valeur d'un paramètre d'un individu et appel à plusieurs évènements SINGLE à partir du nom d'un individu

1. Reprendre le cas 14 et
2. Mettre dans le script de "single2"

```
print("valeur de C1 :", ibm:getModelValue("Loup_0", "C1"))
```

3. Simuler, remarquer le changement de la valeur C1 de "Loup\_0" à partir de la date 50.

### Cas 16 : Obtenir le nom d'un individu

1. Ouvrir le test\_0.vpz avec le plugin IBM
2. Nettoyer les évènements SINGLE des cas précédents, suppression de single2 (sélectionner la ligne de single2 et cliquer sur "-") et des lignes `ibm:addEvent("SINGLE",49,"single2");` et `ibm:addEvent("SINGLE",51,"single2");`
3. Mettre dans single

```
print(ibm:getModelName("Loup", 4))
```

4. Simuler, remarquer la trace dans la console à la date 50, c'est le nom du 4ème Loup.

### Cas 17 : Obtenir le nom de tous les individus

1. Ouvrir le test\_0.vpz avec le plugin IBM
2. Mettre dans single le script qui itère tous les individus Loup, `ibm:getNumber` renvoie le nombre de loup vivants

```
for i=1,ibm:getNumber("Loup"), 1 do
    print(ibm:getModelName("Loup", i))
end
```

3. Simuler, remarquer la trace dans la console à la date 50, nom de tous les loups vivants.

### Cas 18 : Supprimer un individu

1. Ouvrir le test\_0.vpz avec le plugin IBM
2. Mettre dans single

```
ibm:delModel("Loup_0")
```

3. Simuler, remarquer la mort de l'individu nommé "Loup\_0" dans les observables

## Cas 19 : Évènements réguliers

1. Ouvrir le test\_0.vpz avec le plugin IBM
2. Mettre dans single

```
print(ibm:getTime())
```

3. Mettre dans le script principal

```
ibm:addEvent("INIT","init")
ibm:addEvent("RECUR",10,2,"single");
```

4. Simuler, remarquer la trace dans la console

## Cas 20 : Observer une variable lua

1. Ouvrir le test\_0.vpz avec le plugin IBM
2. Mettre dans single

```
acc = 0
for i=1,ibm:getNumber("Loup"),1 do
    acc = acc + ibm:getModelValue("Loup",i,"C1");
end
```

3. Mettre dans le script principal

```
ibm:addEvent("INIT","init")
ibm:addEvent("RECUR",10,1,"single");
```

4. Laisser init du Cas 9. Valider.
5. Ouvrir Atomic Model du Controleur
6. Aller dans l'onglet Observables et ajouter obs\_Controlleur
7. Editer obs\_Controlleur
  - (7.1) Cliquez-droit dans la colonne Observable, add
  - (7.2) Entrer le nom "acc" de notre nom de variable. Valider.
  - (7.3) Ajouter la vue test\_0View à acc. Valider.
8. Cocher le bouton radio de obs\_Controlleur
9. Enregister, simuler

### **Cas 21 : Se servir d'un évènement comme fonction**

1. Ouvrir le test\_0.vpz avec le plugin IBM
2. Diminuer le nombre d'individus créés dans init à 2 ou 1
3. Mettre dans le single init (), ce qui appellera la fonction init
4. Simuler