

PRISE EN MAIN DE L'IDE

Attention, ce document est un résumé en français pour OMNeT++ 5.6.1 se basant sur le document fourni par OMNeT++ [User Guide version 6.0.1](#).

J'ai par la suite trouvé le [User Guide version 5.6](#). Document disponible dans la partie « Anciennes documentations » d'OMNeT++.

Toutes les informations présentes dans ce document ont été testées dans OMNeT++ 5.6.1.

Ce document est voué à être complété/modifié.

Si des informations sont incorrectes ou manquantes n'hésitez pas à me contacter.

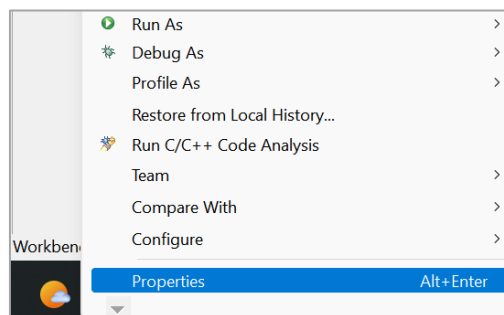
I. Généralité

Workspace : répertoire où se trouve des projets.

Pour changer de Workspace / d'espace de travail: **file >> Switch Workspace >> Other**.

Projet : « dossier » pour ranger les fichiers. **File >> New >> OMNeT++ Project**.

- Configuration du projet : **Project Explorer >> Properties**.



- Un projet peut référencer un autre projet dans le même Workspace. Cela est fait dans **Project References**.

Le projet inclut la prise en charge de l'édition de fichiers NED, MSG et INI, ainsi que le développement C++ de modules simples. Il faut :

- Choisir un Template (vide ou un existant)
- Choisir une Toolchain (souvent il n'en existe une unique donc la sélectionner)
- Choisir une configuration (pour le débogue)

II. Fichiers NED

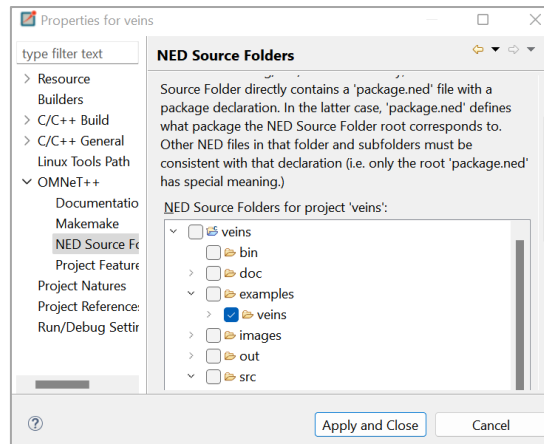
1. Créer un fichier NED

Pour créer un nouveau fichier ned : **File >> New >> Network Description File**.

Il est important que le fichier ned ait le même nom que le module qui le contient.

Il est possible de spécifier à l'IDE le dossier dans lequel il trouvera les fichiers NED :

Project Explorer >> Properties (Alt + enter) >> OMNeT++ >> NED Source Folders.



2. Editeur NED

Dans l'IDE double cliquer sur un fichier « .ned ». L'éditeur NED se lance. Si la représentation graphique ne se lance pas la version texte s'affiche. Les raisons d'un échec du mode graphiques sont entre autres :

- Une erreur de code
- La localisation du fichier. Il ne se trouve pas dans le dossier source NED.

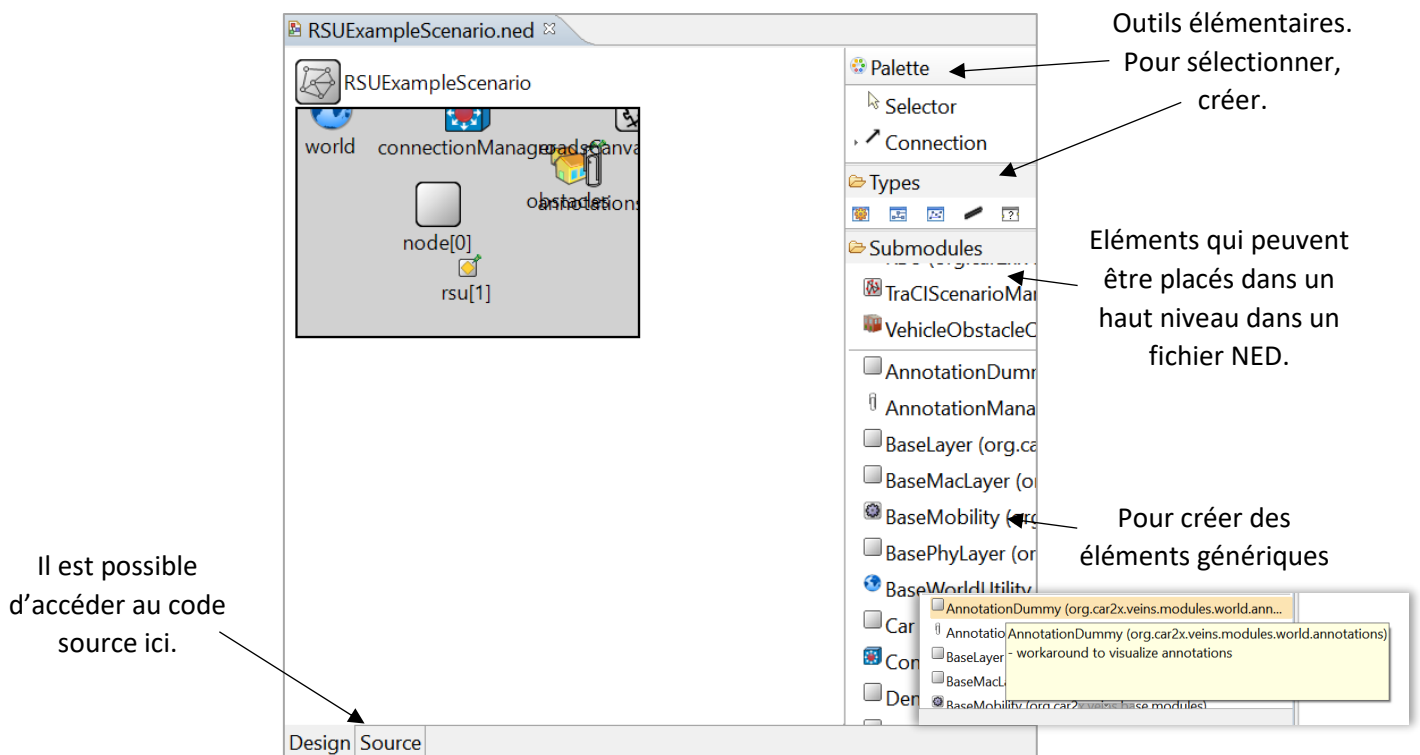
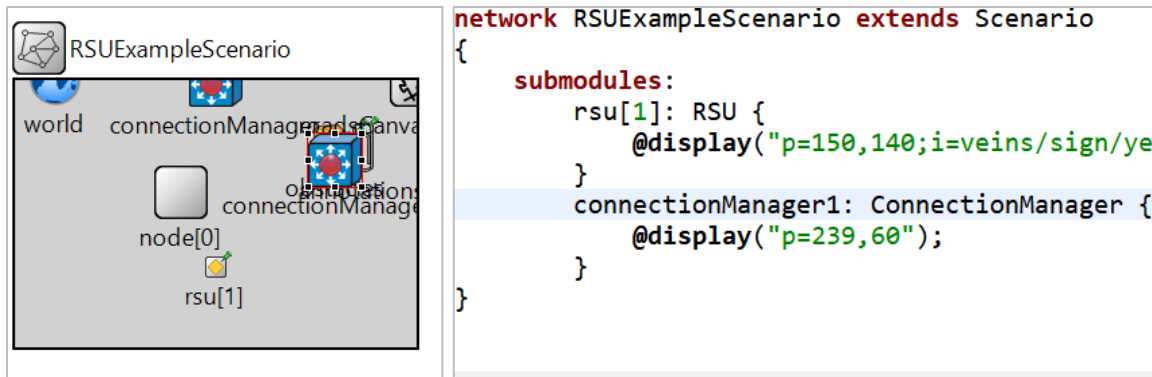


FIGURE 1 EXEMPLE DE REPRESENTATION GRAPHIQUE D'UN FICHIER NED

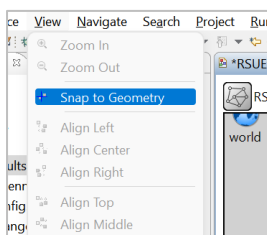
Quand on clique sur un élément graphique, le code source nous le retrouve automatiquement.



a. Mode graphique

- Modules simples et sous-modules :
 - o Représentés sous forme d'icônes.
- Modules composés et réseaux :
 - o Représentés sous forme de rectangles.
- Connexions entre les modules / sous modules :
 - o Représentées par des lignes : connexion bidirectionnelle.
 - o Représentées par des flèches : connexion unidirectionnelle.

Pour créer un module il suffit de cliquer sur le module souhaité et de le déposer là où on le souhaite. Les sous modules ne peuvent être placé qu'à l'intérieur de modules composés ou de réseaux.

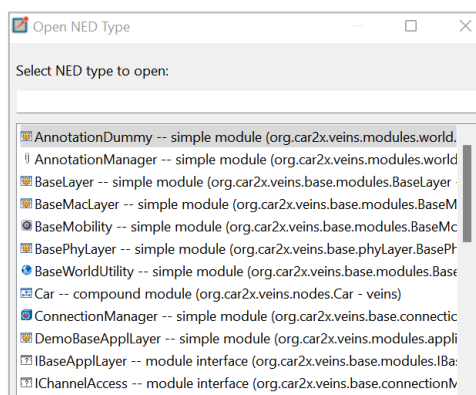


Dans le menu View l'option Snap to Geometry permet d'afficher des lignes pour aligner les modules.

Pour changer les propriétés d'un module :

- Cliquez droit (ou Ctrl + entrer) sur le module en mode graphique
- Sélectionner Properties.

Pour trouver un module par son type Ctrl+Shift+N :



b. Mode texte

- Pour afficher la liste de tous les raccourcis clavier utile : Ctrl+Shift+L.

- **Ctrl+SPACE** : L'éditeur proposera des mots ou des modèles possibles. Ceci est sensible au contexte, il n'offrira donc que des suggestions valides.
- Nettoyer et réparer les fichiers NED : **Project >> Clean Up NED Files**.

III. Fichiers INI

Dans OMNeT++, les modèles de simulation sont paramétrés et configurés pour être exécutés à l'aide de fichiers de configuration portant l'extension .ini, appelés fichier INI.

1. Créer un fichier INI

Pour créer un nouveau fichier ned : **File >> New >> Initialization File**.

2. Editeur INI

a. Mode source

Ce mode fournit un éditeur de texte qui permet la mise en surbrillance et la saisie semi-automatique des noms.

- **Ctrl+SPACE** : L'éditeur proposera des mots ou des modèles possibles. Ceci est sensible au contexte, il n'offrira donc que des suggestions valides.

b. Mode Form

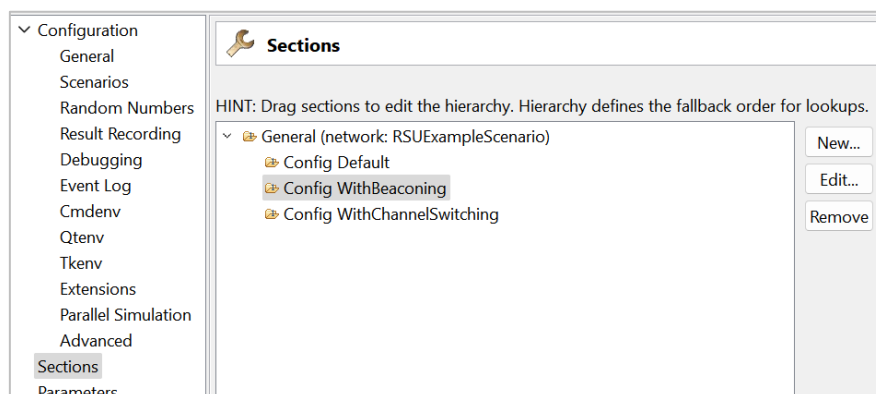
Ce mode permet de modifier la configuration en entrant les valeurs dans un formulaire.

Il est possible de créer plusieurs configurations dans le même fichier INI et y faire référence avec l'option -c lors du démarrage de la simulation.

i. Sections

Les sections sont affichées sous la forme d'un arbre. Les nœuds héritent des paramètres de leurs parents.

Le logo dossier est différent en fonction de ce que contient la section (contient qu'un run, plusieurs répliques...).

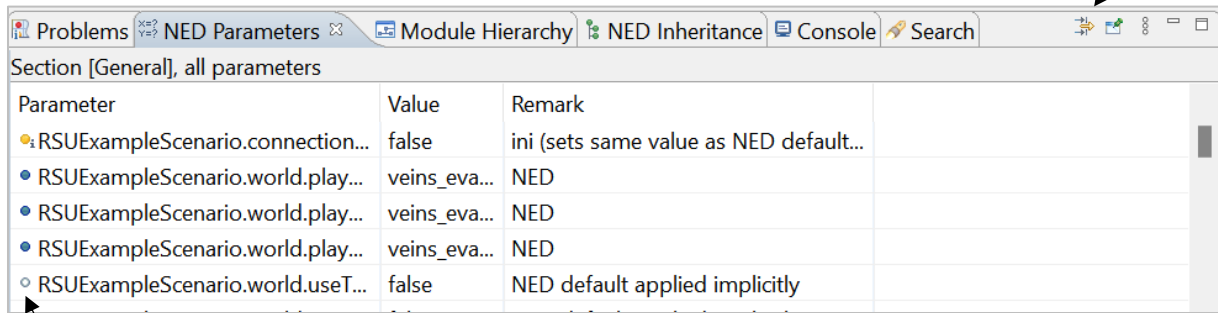


ii. Parameters

Cet onglet permet de paramétrer certains paramètres des modules. Le reste est à paramétrer dans les onglets dossiers comme (Random Numbers, General...

Permet d'afficher
seulement les paramètres
non assignés.

c. Paramètres NED



Parameter	Value	Remark
RSUExampleScenario.connection...	false	ini (sets same value as NED default...
RSUExampleScenario.world.play...	veins_eva...	NED
RSUExampleScenario.world.play...	veins_eva...	NED
RSUExampleScenario.world.play...	veins_eva...	NED
RSUExampleScenario.world.useT...	false	NED default applied implicitly

<input checked="" type="radio"/>	La valeur est définie dans le fichier NED.
<input type="radio"/>	La valeur par défaut du fichier NED est explicitement donnée dans le fichier INI.
<input type="radio"/>	La valeur par défaut du fichier NED est automatiquement appliquée puisqu'aucune valeur n'est spécifiée dans le fichier INI.
<input checked="" type="radio"/>	La valeur est définie dans le fichier INI (peut réécrire sur la valeur du fichier NED)
<input checked="" type="radio"/>	La valeur est définie dans le fichier INI (même valeur que dans le fichier NED)
<input checked="" type="radio"/>	Demande à l'utilisateur pendant l'exécution.
<input type="radio"/>	Pas de valeurs dans le fichier INI et NED.

IV. Fichiers de message

1. Créer un fichier Message (.msg)

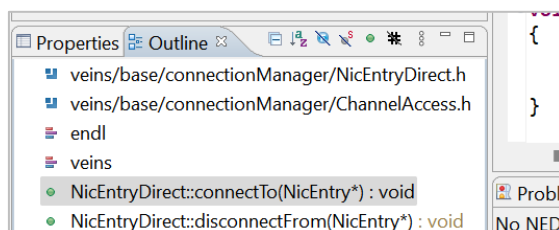
Pour créer un nouveau fichier ned : **File >> New >> Message Definition (msg)**

V. Développer en C++

Il est recommandé de lire la documentation présente dans **Help >> Help Content**.

Les Makefiles sont automatiquement générés avec l'IDE OMNeT++.

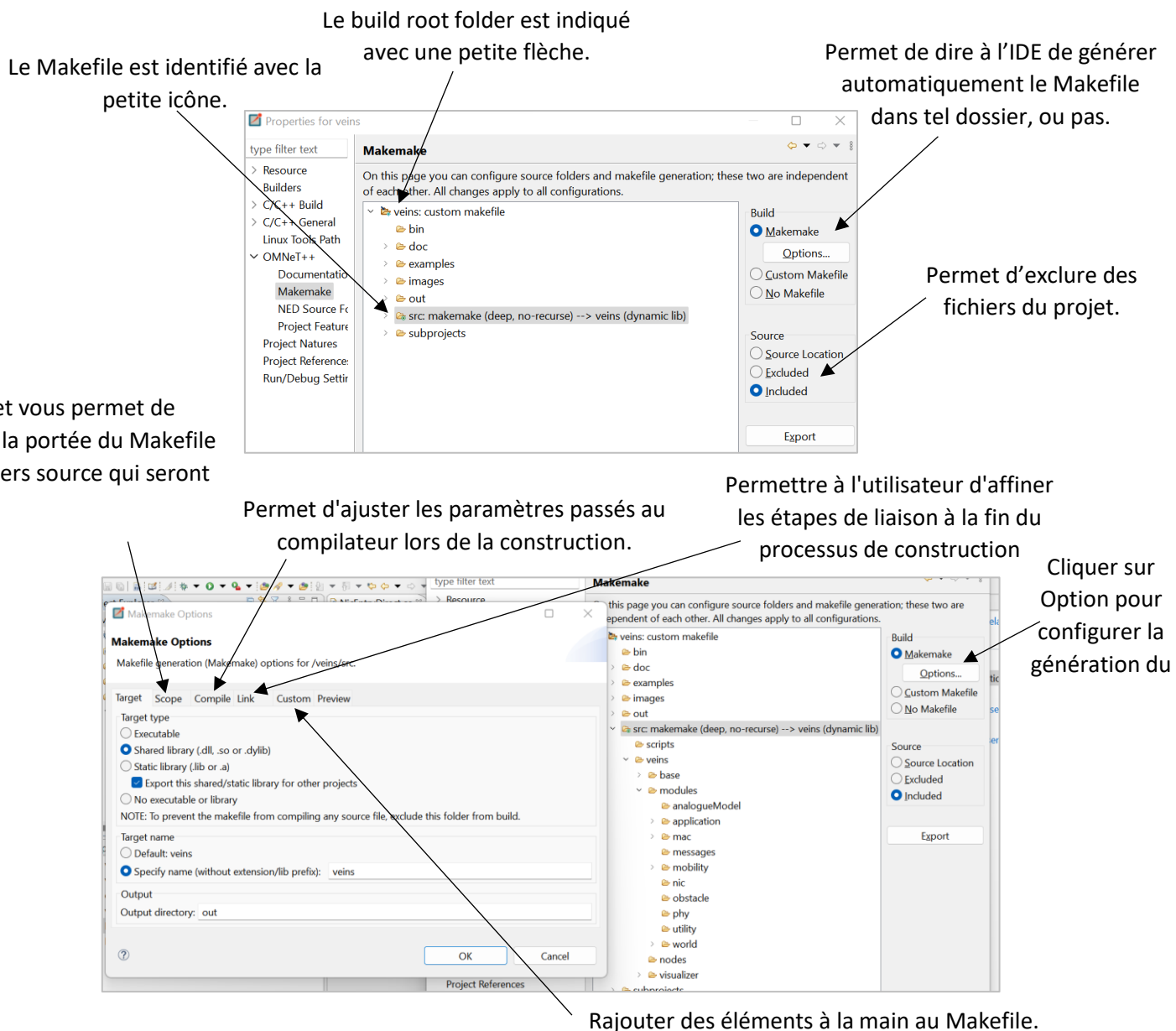
La fenêtre en bas à gauche, onglet **Outline** donne un aperçu de la structure des fichiers source et peut être utilisée pour naviguer rapidement dans le fichier.



Pour Build un projet **Project >> Build Project** ou **Ctrl+B**.

1. Configurer le Build System

Clique droit sur le projet >> **Properties >> Makemake**.



2. Feature

Les features peuvent être définies par projet. C'est une partie du projet (codebase) qui peut être désactivée dans son ensemble, c'est-à-dire exclue des sources C++ (et donc de la construction) et aussi de NED. Les utilisateurs finaux n'ont que la possibilité d'activer/désactiver ces fonctionnalités.

Pour activer ou désactiver des Features : **Project >> Project Features**.

L'activation des fonctionnalités sont sauvegardés dans le fichier « .oppfeaturestate » dans le dossier « root ».

a. Ligne de commande

Il est possible de la faire également en ligne de commande (opp_makemake, ect.) avec l'aide de la commande opp_featuretool.

Pour plus d'information cf. p.51.

b. Fichier .oppfeatures

Les fonctionnalités du projet sont définies dans le fichier .oppfeatures du répertoire racine du projet. Ce fichier, entre autres, n'est pas visible dans l'explorateur de fichier depuis l'IDE d'OMNeT++.

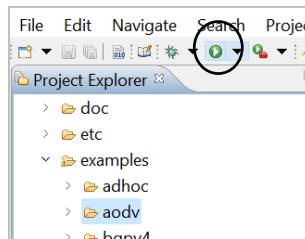
c. Introduire une Feature à son projet

- 1- Isoler le code qui implémente la Feature. Mettre le code dans un dossier à part.
- 2- Vérifier le reste du projet. Faire attention aux référencements vers la Feature. Si référencement, utiliser la compilation conditionnelle (`#ifdef WITH_YOURFEATURE`).
- 3- Rajouter la description de la Feature dans le fichier `.oppfeatures`.
- 4- Tester.

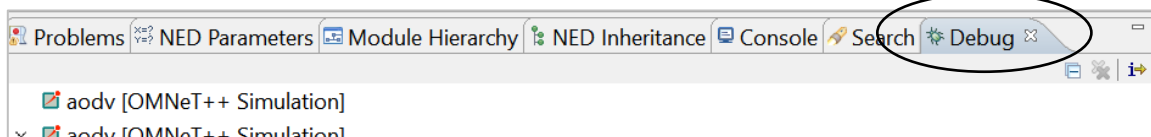
VI. Exécution

Pour lancer un programme il existe deux méthodes :

- En cliquant sur Run As directement sur le fichier :
 - o `Nom_projet >> sous_dossier_si_besoin >> nom_simu.ini`
 - o Cliquez droit >> Run As >> OMNeT++ Simulation
- En cliquant sur le logo « run » directement :
 - o Se placer sur le dossier
 - o Cliquer sur le logo « run », flèche lecture verte/blanche en haut.



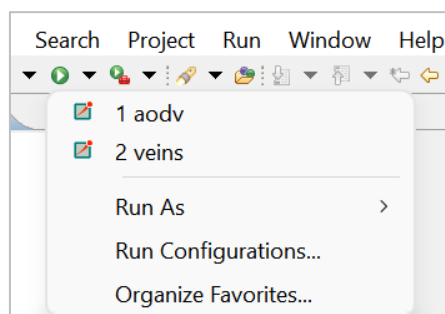
Il est possible de voir les simulations lancées dans la Debug view. Pour afficher cet onglet cliquer dans le menu sur `Window >> Show View >> Other >> Debug >> Debug`. Cela rajoute un onglet dans la fenêtre du bas avec les simulations lancées.



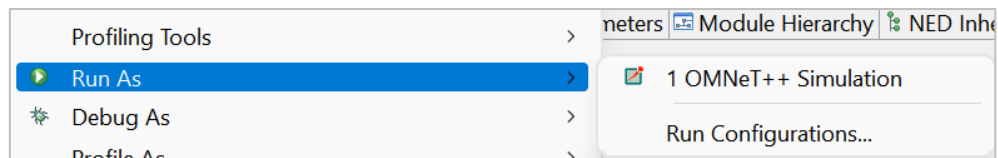
1. Configurations du Run / de l'exécution

Plusieurs manières existent pour ouvrir la fenêtre « Run Configurations » :

- Dans la barre du menu :
 - o Cliquer sur la flèche noire à droite de l'icône « run ».
 - o Cliquer sur « Run Configurations »



- Directement sur Run As :



Cette fenêtre de configuration permet :

- Rentrer le nom de l'exécutable de la simulation.
- Spécifier le working directory du programme.
- Spécifier les fichiers INI (d'initialisation)
- Spécifier quel run doit être exécuté.
- Choisir le nombre de CPU
- ...

2. Pretty printers

De nombreux programmes contiennent des structures de données dont le contenu est difficile à comprendre (variables "brutes" du programme). Les pretty printers sont des classes Python que gdb invoque pour transformer une structure de données réelle en quelque chose de plus facile pour l'humain à comprendre. Les fichiers *.py qui fournissent et enregistrent ces pretty printers sont généralement chargés via le script de démarrage de gdb, .gdbinit.

Ce fichier est présent dans : C:\Users\nom_user\source\omnetpp-5.6.2\misc\gdb.

Certain pretty printers peuvent occasionnellement interférer avec le programme en mode débogue, il peut donc parfois être utile d'arrêter temporairement les pretty printers. Pour ce faire entrer la commande suivante : `disable pretty-printer global omnetpp;.*`

VII. QTENV : exécution graphique

Timeline : Affiche le contenu de l'ensemble des événements futurs (FES) sur une échelle de temps logarithmique.

Top Status : affiche soit des informations sur le prochain événement de simulation (en mode Step et Normal), ou des données de performance (comme le nombre des événements traités par seconde (en mode Rapide et Express)).

Gestion de la vitesse de la

Object Navigator : affiche la hiérarchie des objets dans la simulation actuelle et dans le FES.

Object Inspector : Affiche le contenu et les propriétés de l'objet sélectionné.

Network display : Affiche graphiquement le réseau ou n'importe quel module. C'est aussi là que se déroule l'animation.

Log Viewer : Affiche le journal des paquets ou des messages envoyés entre les modules, ou le journal des messages émis par les modules pendant la simulation. Les N derniers événements sont gardés (ce paramètre peut être changé dans File >> Preferences* >> Logs >> Overall history size). Il y a deux modes :

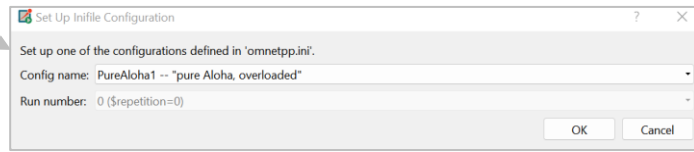
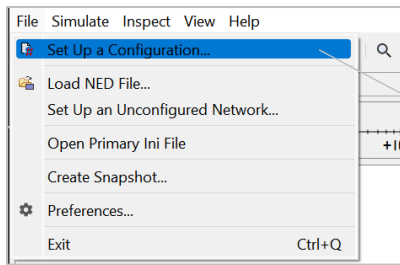
- **Log Viewer** : affiche le contenu lié au module inspecté dans l'affichage réseau au-dessus à tout moment (affiché sur l'image du dessus)
- **Message Mode** : affiche les messages envoyés entre les sous-

Event#	Time	Relevant Hops	Name	ID	Kind	Length
#1	0.223892321265	host[15] --> server	pk-18-#0	21	0	119 bytes
#3	0.251748818876	host[6] --> server	pk-9-#0	23	0	119 bytes
#5	0.309459361	host[10] --> server	pk-13-#0	25	0	119 bytes

1. Lancer et paramétrer QTENV

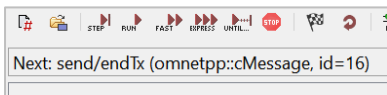
Au démarrage, Qtenv lit le ou les fichiers INI spécifiés sur la ligne de commande (ou omnetpp.ini si aucun n'est spécifié), et configure automatiquement la simulation qui y est décrite. S'ils contiennent plusieurs configurations de simulation, Qtenv demande laquelle mettre en place.

Pour changer de configuration à exécuter : *Set up a configuration.*



Les différents types d'exécution de la simulation :

Step run : Pour lancer seulement le prochain événement. L'ID l'événement suivant est disponible dans le *Top Status* de gauche.



Il est également possible de passer par le menu : **Simulate >> One Step**.



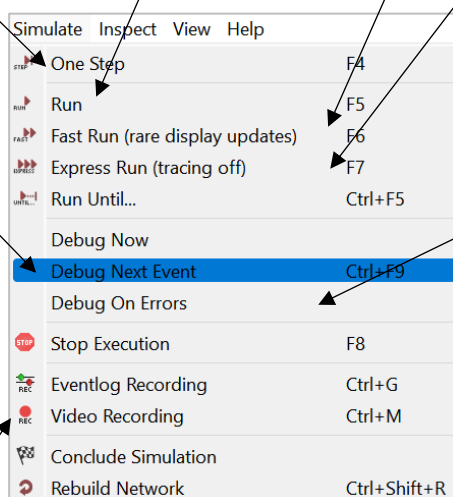
Run : l'animation des messages est active, les fenêtres sont constamment mises à jour. Les messages de sortie sont affichés dans la fenêtre principale. Il est possible d'interagir pleinement avec l'interface utilisateur pendant l'exécution de la simulation (par exemple, ouvrir l'inspecteur, etc.).

Fast run : l'animation des messages est désactivée. L'inspecteur est mis à jour beaucoup moins régulièrement. Vitesse 10X supérieure à celle du mode **run**.

Express run : la totalité du suivi est désactivée. Il n'y a pas de log.

Run until : lancer la simulation jusqu'à un certain temps/événement/message. Utile pour débogueur.

Debug next Event : effectue un événement de simulation comme **Step**, mais exécute un point d'arrêt juste avant d'entrer dans le code de gestion de l'événement du module. Cela permet d'examiner les variables d'état, de faire du pas à pas etc.



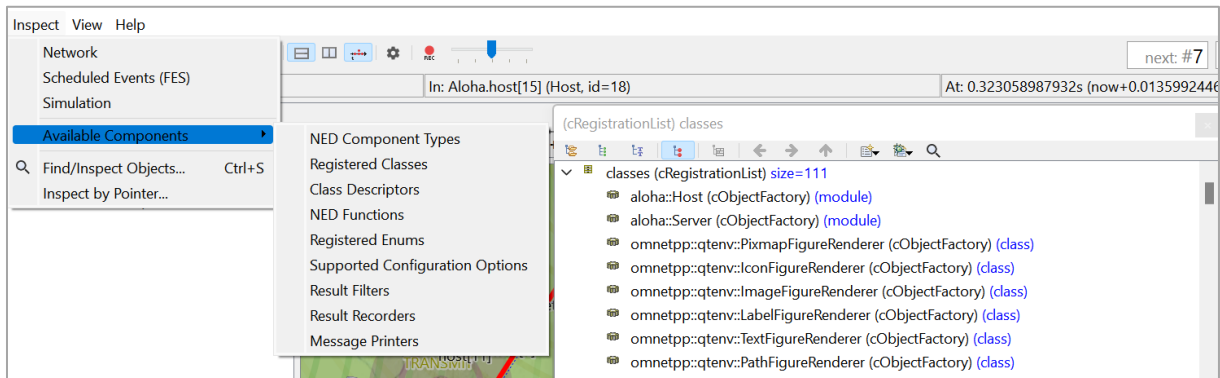
Debug On Errors : Permet de cocher l'option permettant de passer en débogue quand une erreur est repérée pendant l'exécution.

Enregistrer une vidéo : Permet d'enregistrer le contenu de la fenêtre principale (du milieu) dans un sous dossier nommé **frames** qui se trouve dans le working directory. Chaque « frame » est une image PNG. Pour plus d'informations p.72 du document original.

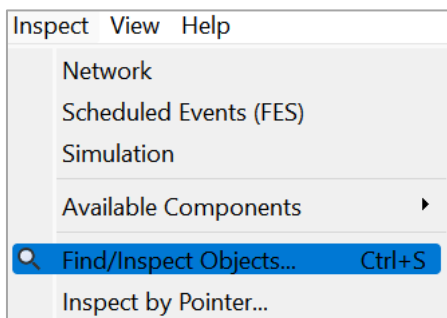


2. Parcourir les composants rentrés

Si un message d'erreur affiche qu'il manque un type, une classe, il est possible de vérifier dans **Inspect >> Available components** si l'objet est disponible.



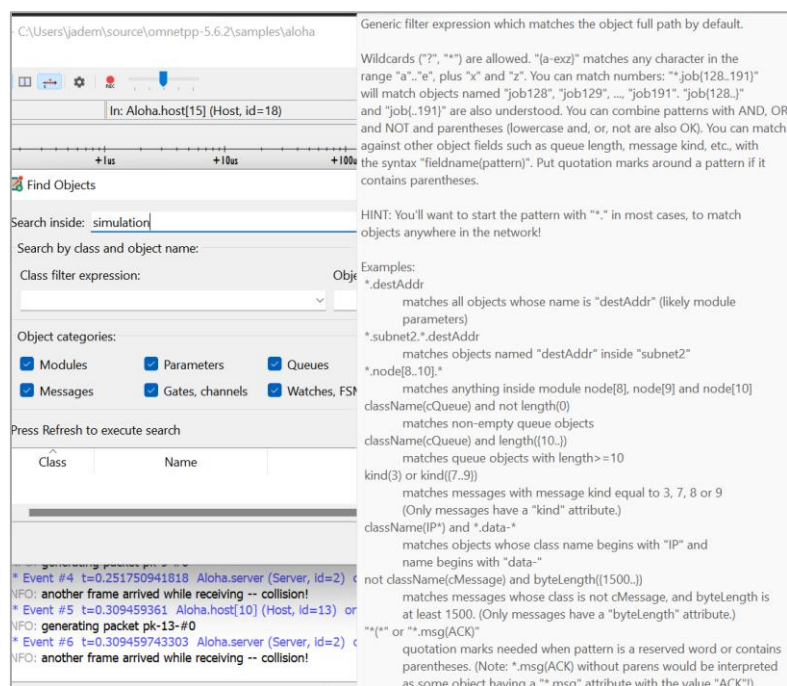
a. Querying Object



Cela permet de rechercher dans la simulation des objets répondant à certains critères. Cela permet entre autres :

- Identifier les routes « embouteillées » dans le réseau en consultant la liste de toutes les files d'attente
 - Rechercher des nœuds avec le plus grand nombre d'abandons de paquets.
 - Trouver les modules qui divulguent les messages.
- Accès facile pour certaines structures de données ou objets, par exemple les tables de routage.

Quand on passe sur le champ « Object full path » une aide s'affiche pour rentrer son filtre :



Exemple pour trouver les longues queues : `length({10..})`

D'autres exemples sont disponibles à la page 86 du document officiel.

3. Fenêtre de préférence

Pour accéder à la fenêtre : **File >> Preferences**.

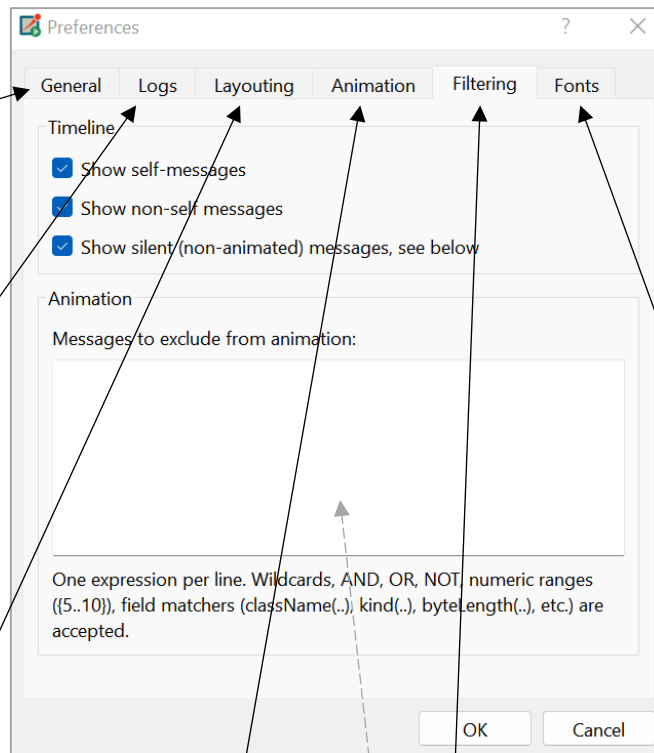
Général : utilisé pour définir le comportement par défaut de l'interface utilisateur.

Exemple : fréquence où l'interface utilisateur sera mise à jour pendant que la simulation s'exécute en mode Express.

Log : pour définir le comportement de journalisation par défaut.

Exemple : le format de préfixe des bannières d'événements.

Layouting : Qtenv fournit une mise en page automatique pour les sous-modules dont les emplacements ne sont pas spécifiés dans les fichiers NED. L'algorithme de mise en page peut être affiné ici.



Animation : Pour affiner la paramétrage des animations. Cette fenêtre permet de désactiver certaines animations. Pour plus de détails cf. p.90 du document officiel.

Filtering : cette page a deux fonctions :

- *Filtrer le contenu de la Timeline* : permet de masquer des messages (self-messages (= timers) et non-animated messages).
- *Supprimer l'animation de message* : *Exemple* supprimer le Data Traffic pour se focaliser sur le routage.

La Text Box permet de spécifier des filtres (un par ligne) des messages par le nom, le nom de la classe etc. Cf. exemples p.90 du document officiel.

Fonts : permet de sélectionner les polices, la taille des différents textes.

4. Lignes de commandes optionnelles

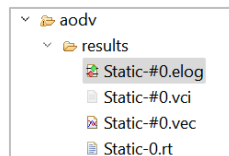
-h	Le programme affiche un message d'aide et sort.
-u Qtenv	Permet au programme de démarrer avec QtEnv.
-f filename	Pour spécifier le nom des documents de configuration. Par défaut : omnetpp.ini
-l filename	Permet de charger des bibliothèques partagées. Possible d'en mettre plusieurs.
-n filepath	Lorsqu'il est présent, remplace la variable d'environnement NEDPATH et définit l'emplacement source des fichiers NED de simulation.
-c configname	Permet de sélectionner une configuration INI pour l'exécution.

Pour plus d'information concernant QTENV et C++ cf. la partie 7.8 et 7.9 du document officiel.

VIII. Sequence Charts (diagramme de séquence) et Eventlog

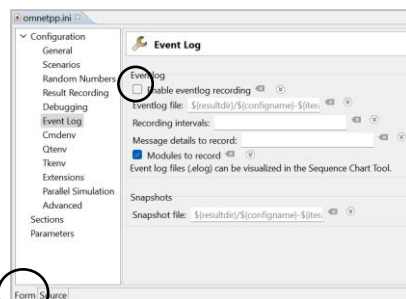
La *sequence chart* et l'*Eventlog* file permet d'afficher le journal des événements enregistré par le noyau de simulation OMNeT++. Les deux outils peuvent afficher des journaux d'événements filtrés

Par défaut, le fichier journal des événements enregistré sera placé dans le répertoire **results** avec le nom **\$configname-\$runnumber.eelog**. Le fichier sera complété quand une simulation est lancée.



L'éditeur de fichier INI dans l'IDE OMNeT++ permet de configurer l'enregistrement automatique du journal d'événements. Pour faire cela :

- Il est possible d'insérer la ligne : **record-eventlog = true** dans le fichier INI.
- De cocher la case dans l'éditeur INI (ouvrir le fichier INI >> Form)



- Ou d'utiliser la ligne de commande « **--record-eventlog** ».
- Ou de cliquer sur le bouton d'enregistrement dans la barre d'outils Qtenv.

Les événements sont enregistrés en fonction de leur temps de simulation et du module où ils se produisent. La clé de configuration "Message details" spécifie ce qui sera enregistré à partir du contenu d'un message. Le contenu du message sera enregistré chaque fois qu'un message sera envoyé.

1. Fichier Eventlog

Le fichier eventlog affiche un journal des événements de manière détaillée. Cet affichage se fait sous forme de tableau. Pour y accéder il faut ouvrir le fichier d'extension « .elog » (cf. juste au-dessus). L'onglet Event log, situé dans la fenêtre du bas, permet d'observer le tableau de log.

Eve...	Time	Details
#0	0s	● Setting up module (omnetpp::Module) AODVNetwork
#0	0s	▶ Simulation started with runId Static-0-20230302-15:59:43-83680
#0	0s	⏏ Keyframe with 0 simulation state events
#0	0s	📦 Creating module (omnetpp::Module) AODVNetwork
#0	0s	💡 Display string changed to bgb=650,650
#0	0s	📦 Creating module (inet::physicallayer::RadioMedium) AODVNetwork.radioMedium under module (or

Il y a deux modes de visualisation :

- **Raw data** : montre exactement ce qu'il y a dans le fichier.

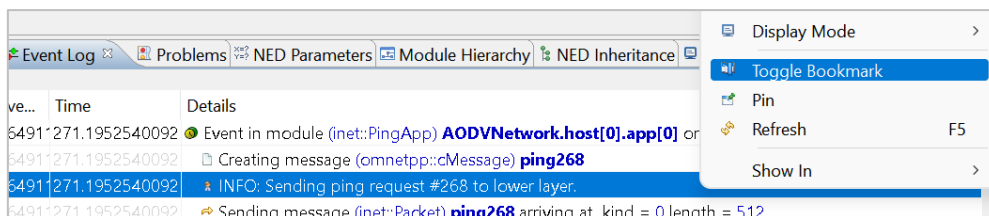


- **Descriptive notation** : affiche les logs dans un format plus compréhensible.



Pour naviguer entre les événements plus rapidement : **Shift+haut** ou **Shift+bas**.

Il est possible de poser des marques pages pour retrouver plus rapidement des événements : clique droit >> **Toggle Bookmark**. Cette fonctionnalité est également disponible avec le mode sequence chart.

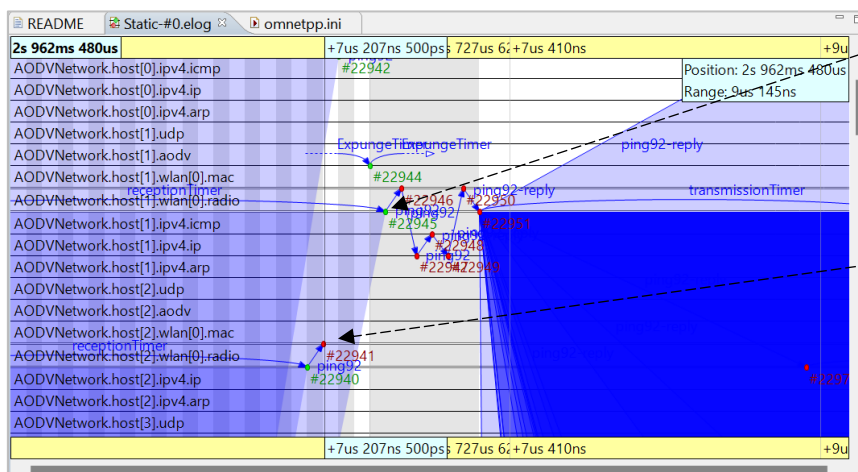


2. Sequence chart

La séquence chart affiche les fichiers du journal des événements sous forme graphique, en se concentrant sur les causes et les conséquences des événements et des envois de messages. Cela permet d'aider l'utilisateur à comprendre les modèles de simulation complexes et à mettre en place les comportements des composants.

Pour se déplacer sur l'axe du temps → clique gauche de la souris et faire glisser.

Axe vertical :
Correspond aux modules de la simulation. Les événements qui se produisent sur un module sont représentés par des

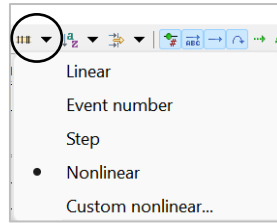


Cercle vert :
représente le traitement d'un « auto-message ».

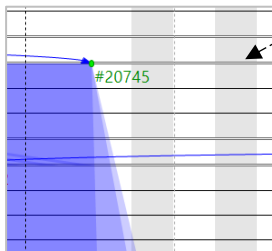
Cercle rouge :
événement causé par la réception d'un message d'un autre module.

FIGURE 2 POUR LA LEGENDE CF. P. 99 DU DOCUMENT OFFICIEL

Ici la représentation est tracée non linéairement. Il existe cependant d'autres manière de représenter les données (linéaire, par étape...).

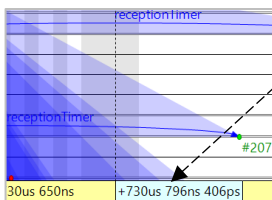


Une région à fond gris indique que le temps de simulation ne change pas le long de l'axe horizontal dans la zone, donc tous les événements à l'intérieur ont le même temps de simulation qui leur est associé.

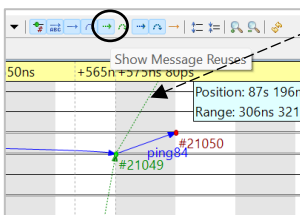


Les modules composés sont représentés par des doubles lignes. Il affichera les événements de tous les modules contenus, à l'exception des événements internes et ceux qui ont leur propre axe d'affiché.

Èvènement interne : un èvènement est interne à un module composé si seulement il traite et envoie les messages aux autres modules qu'il contient.

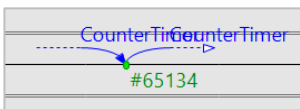


Pour voir le temps de simulation à un point spécifique du graphique, déplacer la souris vers l'endroit souhaité.

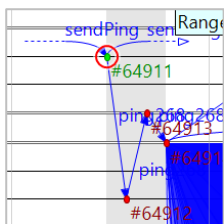


Message reuse : lorsqu'un message arrive dans un module et qu'il se voit stocké pour être renvoyé plus tard. Ceci est représenté par une flèche en pointillés verts entre les deux événements.

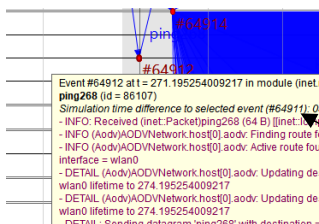
Ces flèches ne sont pas visibles pas défaut. Pour les voir cliquer sur l'icône en haut.



Quand un message va trop loin pour afficher la flèche en entier. Seules les extrémités sont donc représentées.



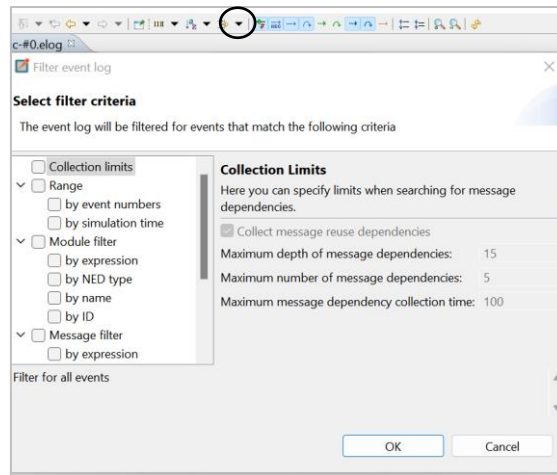
Astuce : cliquer sur l'évènement choisi (l'évènement sera entouré en rouge) et faire **Ctrl+Droite** ou **Ctrl+Gauche** pour aller à la réception ou à l'envoi du message. Cela fonctionne également pour Eventlog.



Quand le curseur est sur un èvènement une info bulle s'affiche dans un cadre beige/jaune.

Si un autre èvènement est sélectionné et que l'info bulle est affichée sur second èvènement, l'info bulle de ce dernier affiche le temps qui sépare ces deux èvènements.

Il est possible de filtrer les messages pour ne pas tous les afficher. Les critères sont configurables en cliquant sur le logo entouré ci-dessous et sélectionnant les critères dans la fenêtre qui s'ouvre par la suite.

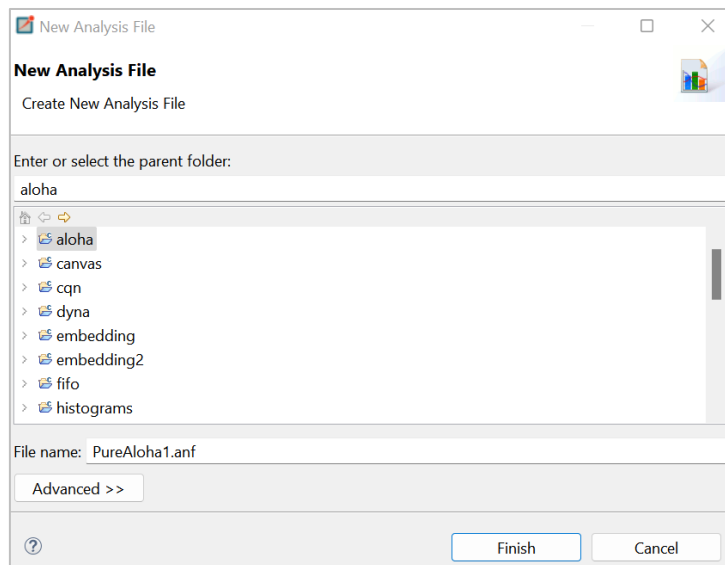


Pour plus d'information sur les filtres cf. partie 8.5 du document officiel.

IX. Analyser les résultats

Attention cette partie est à améliorer. OMNeT++ offre des possibilités d'analyse graphiques intéressantes que je n'ai pas entièrement comprises pour le moment. Pour cette partie se baser sur le document officiel de la version 5.6.2 partie 10.

- Si le fichier existe déjà : double cliquer sur les fichiers d'extension « sca » dans le dossier « results ». Une fenêtre s'affiche :

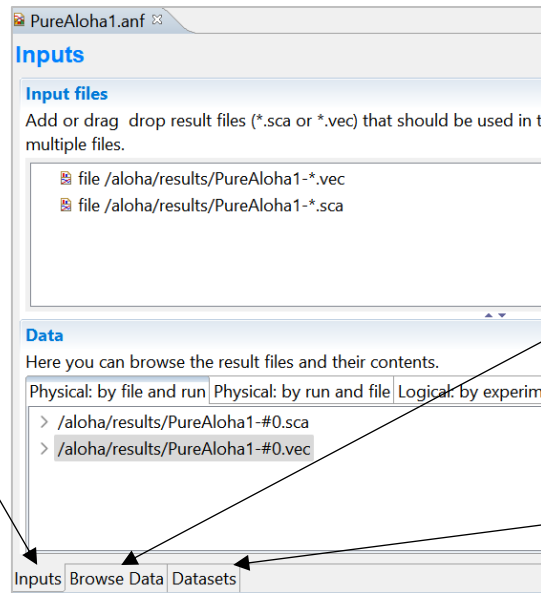


Cliquer sur **Finish**.

- Sinon créer des fichiers d'analyse : se rendre dans **File >> New >> Analysis File >> Finish**. Cependant des fichiers créés ainsi ne comportent pas de références vers le fichier de résultats. Il faut donc les rajouter.

Dans les deux cas cela crée un fichier d'extension « anf ». L'ouvrir permet d'avoir accès à différentes pages :

Input : permet de spécifier les fichiers à analyser.



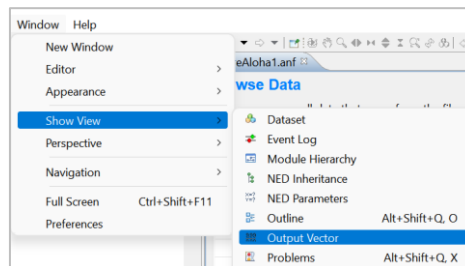
Browse Data : permet d'afficher les résultats (paramètres, vecteurs, diagrammes...)

(Cf. en dessous pour plus de détails)

Datasets: permet de visualiser les graphismes réalisés pendant l'analyse.

1. Voir le contenu sous forme de vecteurs

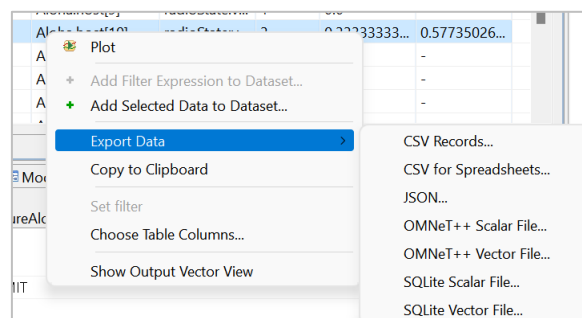
1- Afficher la fenêtre « Output Vector » :



2- Cliquer sur un vecteur. Cela affiche dans la fenêtre préalablement ouverte, en bas dans l'onglet Output Vector, les données :

Item#	Eve...	Time	Value
0	0	0	IDLE
1	2	0.223894237036	TRANSMISSION
2	4	0.251750941818	COLLISION
3	6	0.309459743303	COLLISION
4	10	0.40862640997	IDLE

2. Exporter les données sous différents formats :



3. Générer des graphismes avec Datasets

A compléter et tester.

Datasets

Here you can browse the datasets you have created from the input.

dataset Test

add scalars: all

discard scalars: name("dropped jobs")

add vectors: run(Ring-0-20080610-14:51:02-3276)

group

apply movingavg(alpha=0.1)

line chart line chart

select vectors: name(*length*)

Dataset

Add

Discard

Apply

Compute

Group