Micro Manager for Icy

Plan

Installation
Configuration
Acquisition

- Microscope Snapper
- Live 2D & Live 3D
- Advanced acquisition

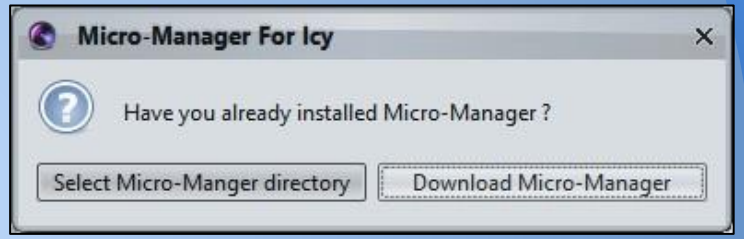
Développement en (Java)Script
Développement de Plugin pour µManager dans Icy

Installation

Par défaut le plugin *µManager for Icy* doit être déjà installé. Verifiez que vous avez bien la dernière version du plugin et lancez le.

Installation

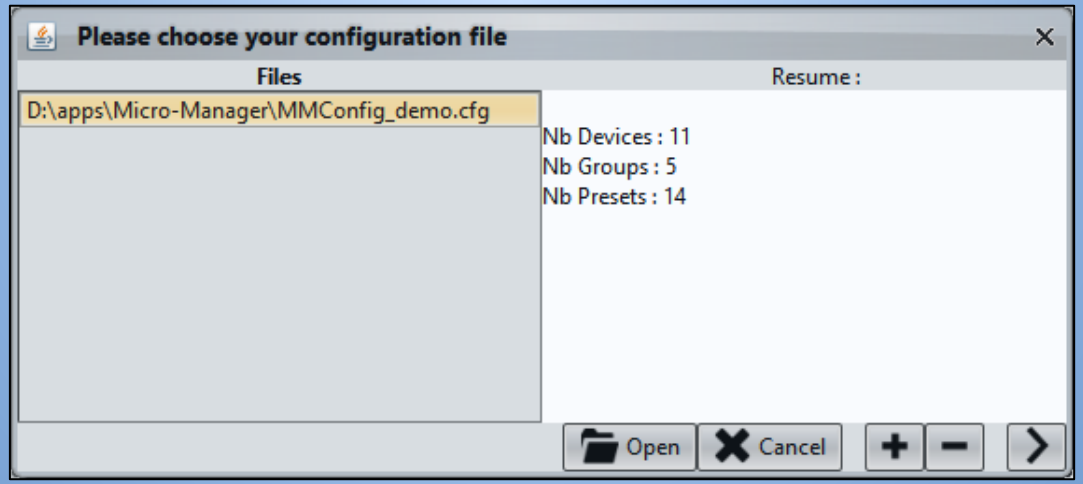
Au premier démarrage vous devez indiquer à Icy où est installé µManager.



Assurez vous que la version installée est bien compatible avec Icy (actuellement seules les versions 1.4.16 jusqu'à 1.4.18 sont supportées).

Configuration

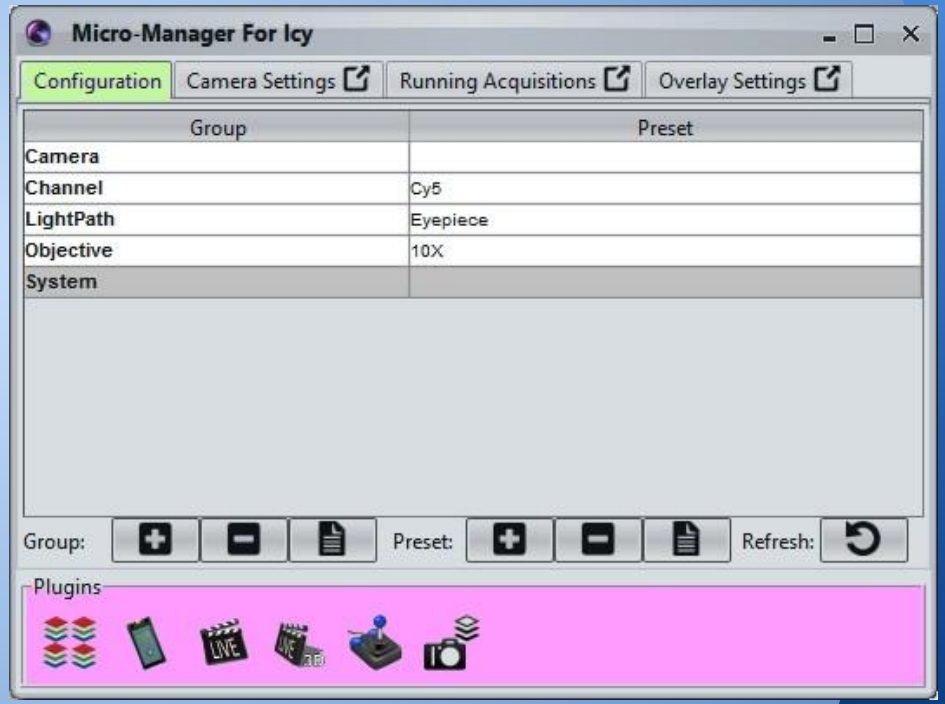
Comme dans µManager vous devez ensuite sélectionner la configuration à charger.



Configuration

La fenêtre principale se compose d’ une partie inférieure (surlignée en rouge ici) dans laquelle on retrouve l’ensemble des plugins compatibles avec *µManager pour Icy*.

La partie supérieure rassemble plusieurs onglets. Le premier onglet *Configuration* reprend ni plus ni moins la partie *Configurations Settings* de µManager avec les *Groups* et les *Presets*.

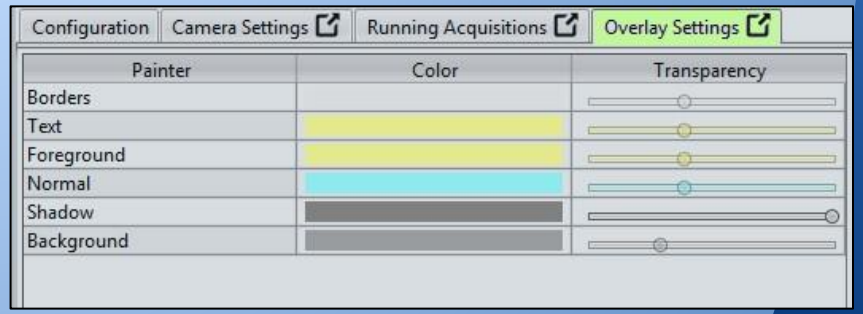


Configuration

Le second onglet reprend les paramètres caméra (Camera settings) de µManager.

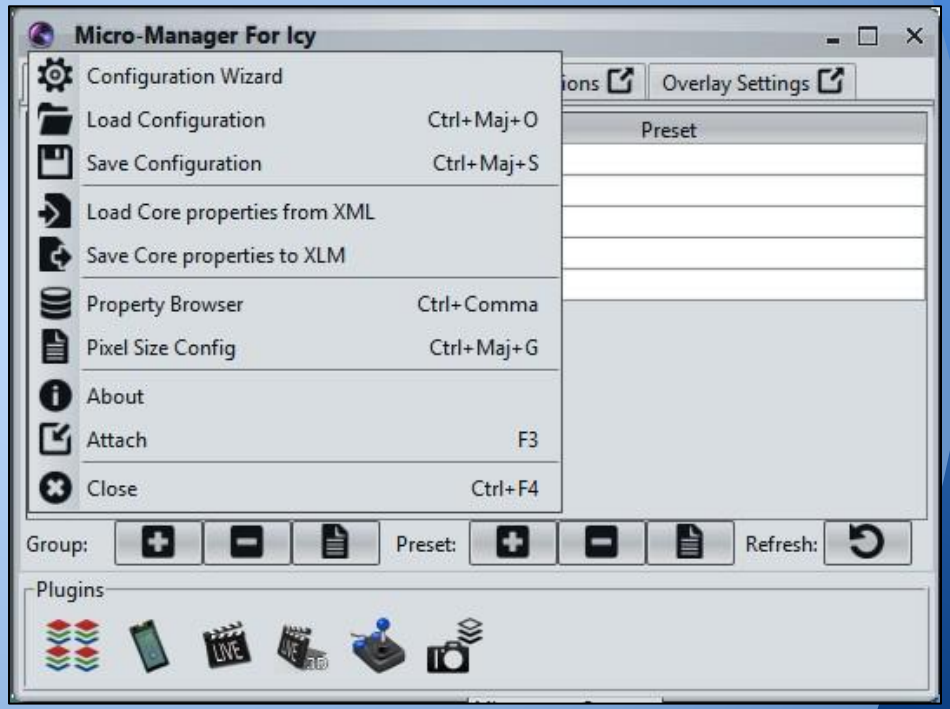
Le 3eme onglet permet d'afficher l' état des acquisitions en cours.

Le dernier onglet permet de modifier l'apparence des informations affichées sur les images durant l' acquisition ou la calibration.



Configuration

Il est possible d'accéder à certaines fonctions de base depuis le menu de la fenêtre principale (en cliquant sur l' icone en haut à gauche). On retrouve ici par exemple le chargement et la sauvegarde des fichiers de configuration, le *Configuration Wizard*, le *Property Browser* et le *Pixel Size Config* de µManager.

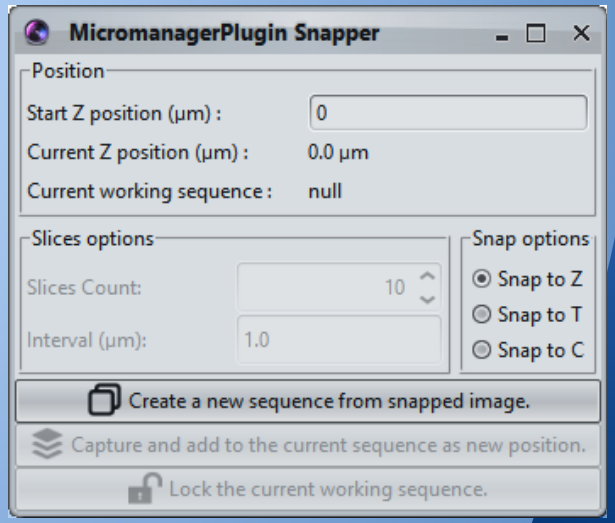


Microscope Snapper

Contrairement au µManager original, toute la partie acquisition est gérée via des plugins séparés. Ainsi l'équivalent des outils *Snap*, *Album*, *Live* ou encore *Multi-D Acquisition* de µManager se retrouvent dans des plugins spécifiques :

Le *Microscope Snapper* rassemble les fonctions *Snap* et *Album* en un seul plugin.

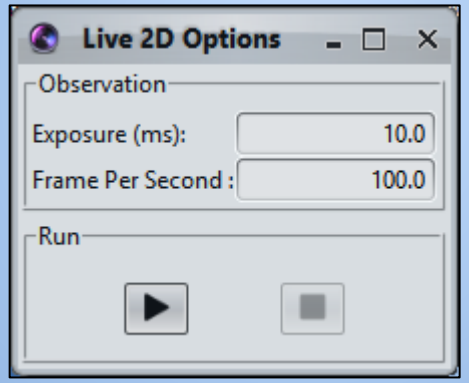
Il permet de faire une acquisition unique ou multiple sur une dimension choisie (Z, T ou C) et de stocker le résultat dans une nouvelle Sequence ou de l'ajouter à une Sequence existante.

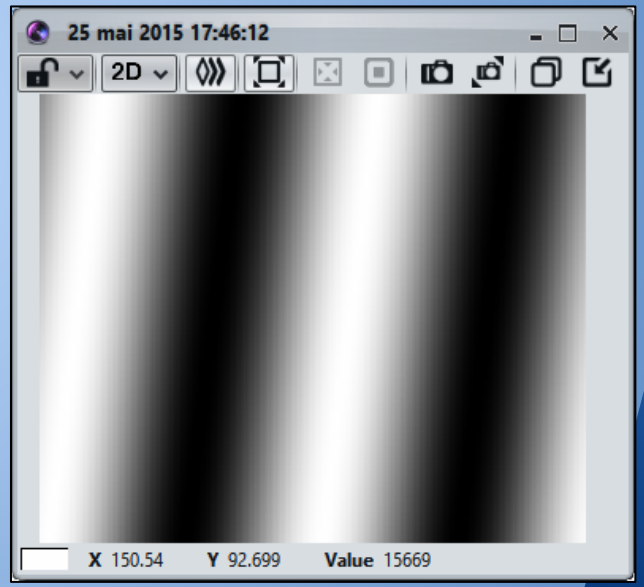


Live 2D

Le Live 2D reprend le même principe que le Live de µManager.

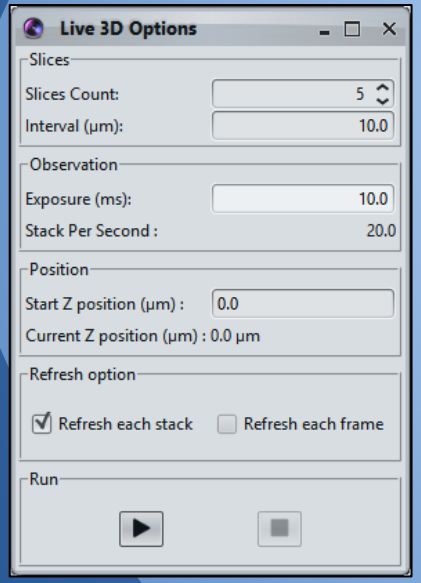
Le plugin donne directement accés au paramètre de temps d' exposition (ou au frame rate désiré) car on souhaite souvent le modifier pour le Live.

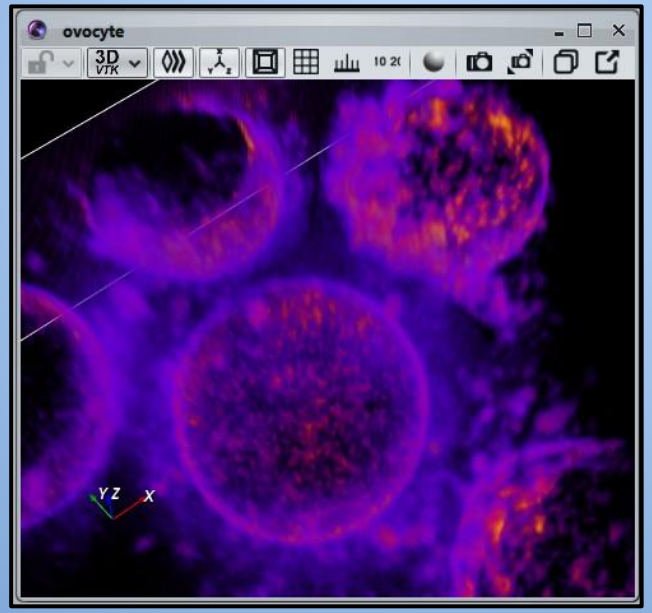


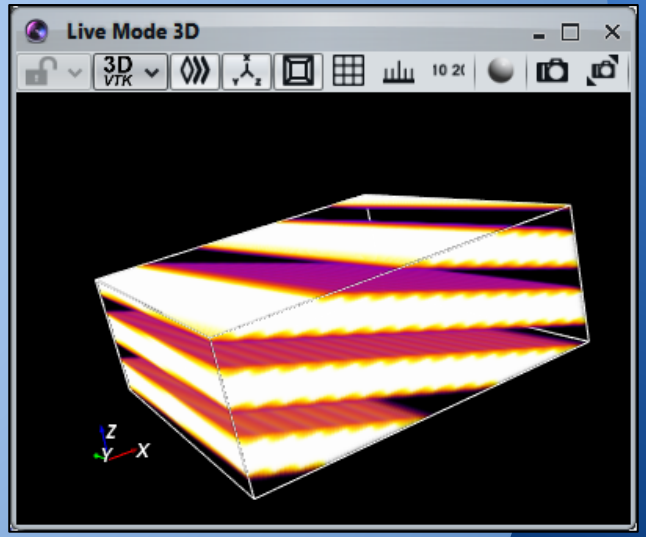


Live 3D

Le Live 3D tire avantage du rendu 3D en raycasting de VTK pour offrir un aperçu temps réel en 3D.



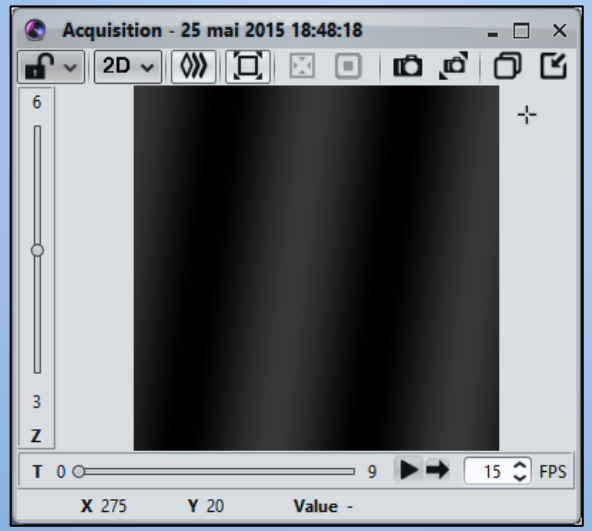


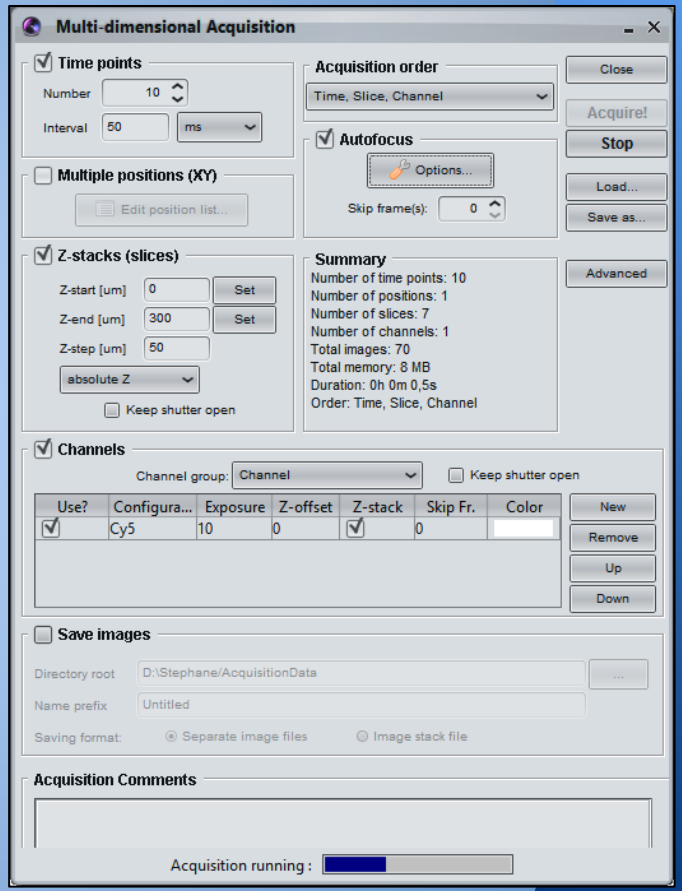


Advanced acquisition

Ce plugin reprend le très puissant *Multi-D acquisition* de µManager.

L'interface est en tout point identique à celle de µManager si ce n'est que l'on peut maintenant voir l'état d'avancement de l'acquisition en cours.

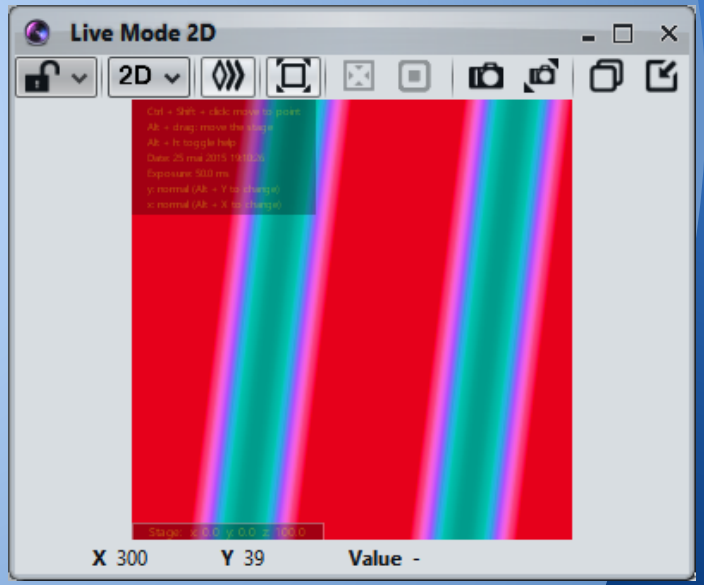




Et les autres...

Le plugin *Remote* permet de controler la position de la platine du microscope sur les axes XY et Z.

Calibrator Manager permet de faciliter la calibration de l'image (calcul de la taille du pixel).



Développement script

MicroManager pour Icy vous permet de controler votre microscope depuis un simple script en java.

L'exemple ci-contre permet d'effectuer les opérations suivantes :

- Déplacer la platine XYZ à la position (5,5,5) µm
- Acquérir une image et l' afficher

```
// move the stage to (5, 5, 5)
StageMover.moveXYAbsolute(5, 5)
StageMover.moveZAbsolute(5)

// acquire a single image
image = MicroManager.snapImage()

// create a sequence and display it
sequence = new Sequence(image)
gui.addSequence(sequence)
```

Script - exercice

Objectifs :

- Positionner le microscope (x,y,z) en (5,5,5)
- Acquérir 3 images
- Déplacer la stage de 10µm en Z entre chaque acquisition
- Afficher l’image dans Icy comme un stack 3D.

Script - exercice - réponse

```
z = 0
sequence = new Sequence(image)          // create the result sequence

StageMover.moveXYAbsolute(5, 5)         // move to position (5, 5, 5)
StageMover.moveZAbsolute(5)

image = MicroManager.snapImage()        // acquire 1 image
sequence.setImage(0, z++, image)        // set it in resulting sequence at position 0
StageMover.moveZRelative(10)            // shift microscope Z position by 10
image = MicroManager.snapImage()        // acquire 1 image
sequence.setImage(0, z++, image)        // set it in resulting sequence at position 1
StageMover.moveZRelative(10)            // shift microscope Z position by 10
image = MicroManager.snapImage()        // acquire 1 image
sequence.setImage(0, z, image)          // set it in resulting sequence at position 2

gui.addSequence(sequence)               // show the sequence in Icy
```

Script - µManager core access

Icy autorise l'accés au core de µManager et ainsi de profiter de l'ensemble des fonctionnalités de l'API de Micro-Manager. Par exemple pour récupérer ou affecter la valeur d'une propriété et plus généralement pour modifier les paramètres d'acquisition (voir Programming Guide - Using device properties)

Utilisation du core sous µManager :

```
core.getProperty(...)
```

Utilisation du core sous Icy :

```
MicroManager.getCore().getProperty(...)
```

Script - µManager core access

```
core = MicroManager.getCore()
image = MicroManager.snapImage()
meta = MicroManager.getMetadata()

println("Binning: " + MDUtils.getBinning(meta))
println("Pixel type: " + MDUtils.getPixelType(meta))

bd = core.getProperty("Camera", "Camera-BitDepth")
exposure = core.getProperty("Camera", "Camera-Exposure")
MicroManager.setExposure(10)
core.setProperty("Camera", "Camera-Binning", 2)
```

Développement plugin

Les principales classes et méthodes à connaitre pour utiliser l'API de µManager dans Icy :

MicroManager	classe principale de µManager pour Icy
StageMover	classe outil pour gérer le positionnement du microscope

MicroManager.getCore()	Permet d'accéder au core de µManager
MicroManager.snapImage()	Acquire une image et la retourne
MicroManager.getMetadata()	Récupère les méta-données de la dernière image acquise
MicroManager.startLiveMode()	Démarre le mode d'acquisition continue (live)
MicroManager.stopLiveMode()	Arrête le mode d'acquisition continue (live)
MicroManager.startAcquisition(...)	Démarre l'acquisition multiple
MicroManager.stopAcquisition(...)	Interrompt l'acquisition multiple
MicroManager.getAcquisitionResult()	Récupère le résultat de l'acquisition multiple

La classe *MicroscopePlugin*

Lorsqu'on développe un plugin Icy pour Micro-Manager il faut utiliser et étendre la classe abstraite *MicroscopePlugin* plutot que *Plugin* ou *PluginActionable*. Dans ce cas il est important de respecter les règles suivantes:

- surcharger la méthode *start()* à la place de la méthode *run()*
- surcharger la méthode *shutdown()* si certaines actions spécifiques doivent être effectuées à la fermeture du plugin.

On est ainsi assuré que µManager sera chargé avant le démarrage du plugin, de plus on peut utiliser les méthodes *onSystemConfigurationLoaded()*, *onCorePropertyChanged()* et *onExposureChanged()* pour détecter les changements de configuration de µManager.

Les évenements

Micro-Manager pour Icy gère plusieurs types d'événements pour faciliter la vie du développeur.

MicroManager.addAcquisitionListener(...)
Permet d'écouter les évenements d'acquisition (start / new image / end).
MicroManager.addLiveListener(...)
Permet d'écouter les évenements du mode live (start / new image / end).

StageMover.addListener(...)
Permet d'écouter les évenements de changement de position du microscope

De cette manière le développeur peut, par exemple, facilement déclencher un traitement spécifique à la réception d'une nouvelle image durant une acquisition.

Plugin - Tutorial project 1

```
public class MyPlugin extends MicroscopePlugin {
	@Override
	public void start()
	 {
		try {
			Sequence result = new Sequence();		// Create the resulting sequence
			StageMover.moveXYAbsolute(5, 5);		// Set microscope X and Y positions
			StageMover.moveZAbsolute(5);			// Set microscope Z position
			result.addImage(MicroManager.snapImage());	// Snap an image and add it to result
			StageMover.moveZRelative(10);			// Move the microscope by 10 µm in Z
			result.addImage(MicroManager.snapImage());	// Snap again
			StageMover.moveZRelative(10);			// Move again
			result.addImage(MicroManager.snapImage());	// Then Snap again
			addSequence(result);				// Finally, show the resulting sequence into Icy
		} catch (Exception e) {
			// Eclipse will ask you to catch the exception, this is caused when we are unable to move the stage
		}
	}
}
```

Plugin - exercice

Objectifs :

- Démarrer le mode *Live*
- S’enregistrer pour recevoir les évenements du *Live*
- Pour chaque image reçue afficher la taille de celle-ci dans la console.

Plugin - exercice - réponse

```
public class MyPlugin extends MicroscopePlugin implements LiveListener {
	public void start() {
		try {
			MicroManager.addLiveListener(this); // register listener first
			MicroManager.startLiveMode();		// then start live acquisition
		} catch (Exception e) {
			// we need to catch possible exception here on startLiveMode()
		}
	}

	public void liveImgReceived(IcyBufferedImage image) {
		try {
			JSONObject meta = MicroManager.getMetadata();
			System.out.println("Image size: " + MDUtils.getHeight(meta()) + " x "  MDUtils.getWidth(meta));
		} catch (JSONException e) {
			// Exception when asked tags doesn't exist
		}
	}

	public void liveStarted() {}
	public void liveStopped() {}

}
```