



## Developers Guide

Georgi Danovski  
e-mail: [georgi\\_danovski@abv.bg](mailto:georgi_danovski@abv.bg)  
(2018)

## Setup

Install your favourite IDK for .NetFramework programming. Choose the program language and create a new Class Library Project. Edit the name of assembly from the Project properties by adding ".CTPlugin". CellTool will recognize as plugins only assemblies with suffix ".CTPlugin.dll". Add a reference to [CellToolDK.dll](#).

The main class of the plugin must have the following construction:

```
using CellToolDK;

public class Main
{
    private Transmitter t;
    private TifFileInfo fi;

    private void ApplyChanges()
    {
        //Apply changes and reload image
        t.ReloadImage();
    }

    public void Input(TifFileInfo fi, Transmitter t)
    {
        this.t = t;
        this.fi = fi;
        //Main entrance
        //You can add your code here
    }
}
```

Public void "Input" is the main entrance of the program. This is the void that will start when the plugin is activated from CellTool menu Plugins and here you must add your code. This void has two arguments - `TifFileInfo fi` and `Transmitter t`. `TifFileInfo fi` contains the data from the active image. The transmitter `t` is used to send back information to CellTool. The command "`t.ReloadImage()`" can be used to send the modified version of the image back to CellTool.

To install the plugin start the "CellTool.exe" and press install button in the Plugins menu. Browse to your assembly and press the "OK" button.

## TiffFileInfo class

Name:	Type:	Description:
roiList	List<ROI>[]	Array that contains one List<ROI> for each channel color
ROICounter	int	Available ROI index. When ROI is added this index must be assigned as ID to it and the value of the variable must be increased by 1
frame	int	Current position in the T stack
zValue	int	Current position in the Z stack
cValue	int	Current position in the C stack
loaded	bool	Returns true if the image loading process is finished
available	bool	Returns true if the image is processed at that time
image8bit	byte[][][]	The image is loaded here if the pixel format is 8 bit - [image index][y][x]*
image8bitFilter	byte[][][]	The processed image is loaded here if the pixel format is 8 bit - [image index][y][x]*
image16bit	ushort[][][]	The image is loaded here if the pixel format is 16 bit - [image index][y][x]*
image16bitFilter	ushort[][][]	The processed image is loaded here if the pixel format is 16 bit - [image index][y][x]*
Dir	string	Directory of the image
imageCount	int	The total number of images in the stack
sizeX	int	X size of the image
sizeY	int	Y size of the image
sizeZ	int	Z size of the image stack
umZ	double	Distance between Z slices in um
umXY	double	Size of one pixel in um
sizeC	int	Number of different color stacks in the image
sizeT	int	Size of the T stack
bitsPerPixel	int	Bits per pixel - value can be 8 or 16
LutList	List<Color>	list with the colors of the different color image stack
FileDescription	string	Description tiff tag (code 270) as a string

\*Image stack dimension order is XYZCT

## ROI class

<b>Name:</b>	<b>Type:</b>	<b>Description:</b>
getID	int	ROI identification number
type	int	Type of the ROI ( 0 - static and 1 - tracking)
Shape	int	Shape of the ROI (0 - rectangle, 1 - oval, 2 - polygon, 3 - freehand, 4 - magic wand)
Width	int	Width for a rectangular and oval ROI
Height	int	Height for a rectangular and oval ROI
Stack	int	Number of ROI layers
D	int	Distance between the ROI layers in pixels
FromT	int	The start position of the ROI in the Time stack
ToT	int	The end position of the ROI in the Time stack
FromZ	int	The start position of the ROI in the Z stack
ToZ	int	The end position of the ROI in the Z stack
GetLocation(int ImageN)	Point[]	Returns an array with one point (upper left angle of oval and rectangular ROI) or an array with points for every angle of a polygonal ROI for the selected image in the image stack
GetLocationAll()	Point[][]	Returns an array with one point (upper left angle of oval and rectangular ROI) or an array with points for every angle of a polygonal ROI for each image in the image stack