A GUIDE TO USING BINARY DATA IN CLICKTEAM FUSION 2.5

MP4   BAT > C:/   DLL   INI

INTRODUCTION

Welcome to this guide on how to use Binary Data inside Clickteam Fusion 2.5. In this PDF, I will guide you through how to insert and use Binary Data. Binary data in your games and applications can prove very useful. It enables you to pack data into your game for external use once running on a users machine. We'll cover more further down.

Welcome to this guide on using Binary Data for Clickteam Fusion 2.5! Some (if not all) of the information in this guide is applicable for Multimedia Fusion 2.0 also. As with all tutorials and guides written in PDF, feel free to display it as a PDF or print to paper. You can study and learn these practices at your own pace, in your own time.

Due to Fusion 2.5's flexible nature of allowing users to accomplish the same result in multiple ways, sometimes there are alternative ways to perform certain routines or functions. These guide(s) will explicitly target the most common, easiest and most effective route.

These hand-written guides are here to help you target or perform a particular routine in Fusion 2.5 enabling you to harness the full potential of what Clickteam Fusion 2.5 can actually do for you as a developer. Whether it's game design or application development you want to do with Fusion 2.5, these guides will hopefully guide you from top to bottom in whatever you're looking to achieve.

This guide has been thoroughly checked before deployment. As always, feedback, comments and suggestions are always welcome on the detail, structure and formatting of these guides. Rather than create individual guides for different learning levels, we hope to be able to reach to all entry level, intermediate and advanced users within the same guide on achieving the same goal.

For more information on how you can download and purchase Clickteam Fusion 2.5, check out the website http://www.Clickteam.com

For more information, tutorials, examples and walk-thrus visit the website.

CONTENTS

For more information, tutorials, examples and walk-thrus visit the website.

1. AN INTRODUCTION TO BINARY DATA

Binary Data. You may ask, what is it? Well, as you may or may not know, data is stored in binary. That's literally 0's and 1's. However, in Fusion it's not as literal as that. Clickteam Fusion enables you to append files to your final application, which in practice means, those files get packed into your final application and can be used as separate files in runtime.
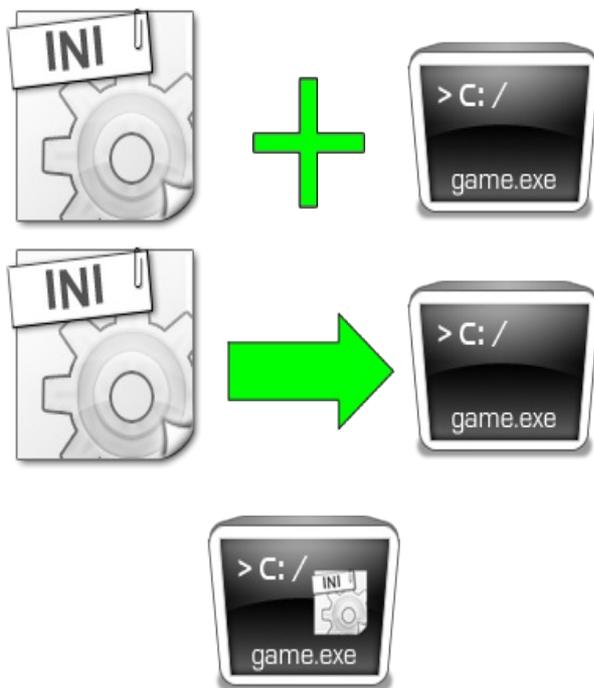
I've draw a quick example here for you. Let's imagine we are building a simple computer game for all platforms. We have created a configuration file that will ship with the game, however when we compile our final build (exe) we want this file to be included with the game itself. So in theory, your final build should be game.exe and config.ini right? That's correct in theory, but we want to INCLUDE this config.ini INSIDE our game.exe so when we distribute our game, we just have one file to distribute.

This is where Binary Data comes into play with Fusion. You can pack virtually any file into your application on your final build, you can then USE this file as if it was a separate file external to your game.

You could literally pack any type of file of any size into Binary. What Fusion does is, on your final build, it will append this binary data to the end of your executable. The file is given a 'link' kind-of index in the final build, so when you 'call' the binary file at runtime, it knows what piece of binary to look for (not in exe – explained below). This is applicable for all runtimes except Windows. When it comes to Windows, you will need to extract the binary manually and then reference the binary filename.

Sound a little bit scary? To a degree, it is. However, once you understand the principal behind the method, it's really easy to use. No need to worry though, as we will cover this entire topic from top to bottom in the proceeding pages.

For more information, tutorials, examples and walk-thrus visit the website.

2. THE DIFFERENCE BETWEEN PLATFORMS

There are two methods of using Binary Data. For Windows applications and games, you will need to manually extract the binary data from the application at runtime and reference it directly.

In all other platforms (*flash, ios, android, xna, html5*) you don't need to do this. This is yet another positive action on behalf of Clickteam's developers, as they don't and wouldn't expect you as the user to waste time fiddling around trying to find out what the temporary folders are for each platform. So Clickteam have implemented a way where you don't have to extract and reference the binary from the application, in the exporters it's literally as easy as just calling the filename.

The reason for the differences between Windows and all the other exporters is because the exporters have different configuration setups for app caches and temporary directories. For example, the temp folder on Xbox 360 could be a deep folder structure as compared to Android which would also differ. However, you maybe wondering why on Windows we can't just reference the filename as we do with the exporters.

The answer to that is quite simply because Clickteam and Fusion 2.5 want to enable you as the developer to have full control over your files and data. It would be fine for Clickteam to adapt Windows to have this feature in but then IF a certain developer did need full control over the extracted binary, it would then mean the developer is now limited. Besides, it's only a subtle difference and we'll cover it in the next topic anyway.

For more information, tutorials, examples and walk-thrus visit the website.

| **Website:** http://www.Clickteam.com – **Community:** http://community.clickteam.com |

3. HOW TO INSERT BINARY DATA

Clickteam Fusion enables you to insert binary files with just a few clicks of the mouse. Open up Clickteam Fusion and locate this icon on the top toolbar

Once you have located this icon, click it. It will open up the DATA ELEMENTS dialog. Data Elements gives you a full rundown of which files are already included in your project. So if you inserted any sounds, music, binary, effects or extensions It will list them all in this dialog.

So for the purpose of this guide, click on the Binary Data tab.

This will list all the 'binary data' we want to include in our app, so let's close this dialog for now and let's go and create an INI file to include as Binary. Click on the Start Menu and open Notepad. Let's create a blank file that looks like this:

> [testgroup]
>
> testnumber=101

Once you have done this, click File > Save As... in the notepad dialog, in the 'Save As Type' dropdown list, select All Files. Now save the file to c:\test.ini

Jump back into Fusion and Click on the Data Elements button (*above*) and click on the Binary Data Tab. Now click on 'Add' and navigate to c:\ and select test.ini

This should have inserted the INI file into the Binary Data tab, you should see it appear in the list like so
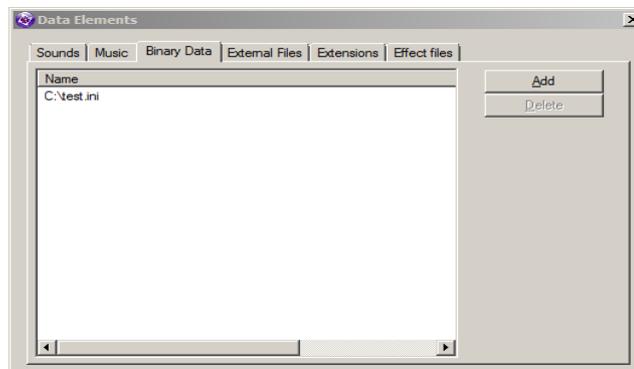
3. HOW TO INSERT BINARY DATA

Once the file is inserted, you can just close the dialog. Fusion will only 'include' this file upon compiling your application, not before.

Now, let's drop an INI object into our frame.
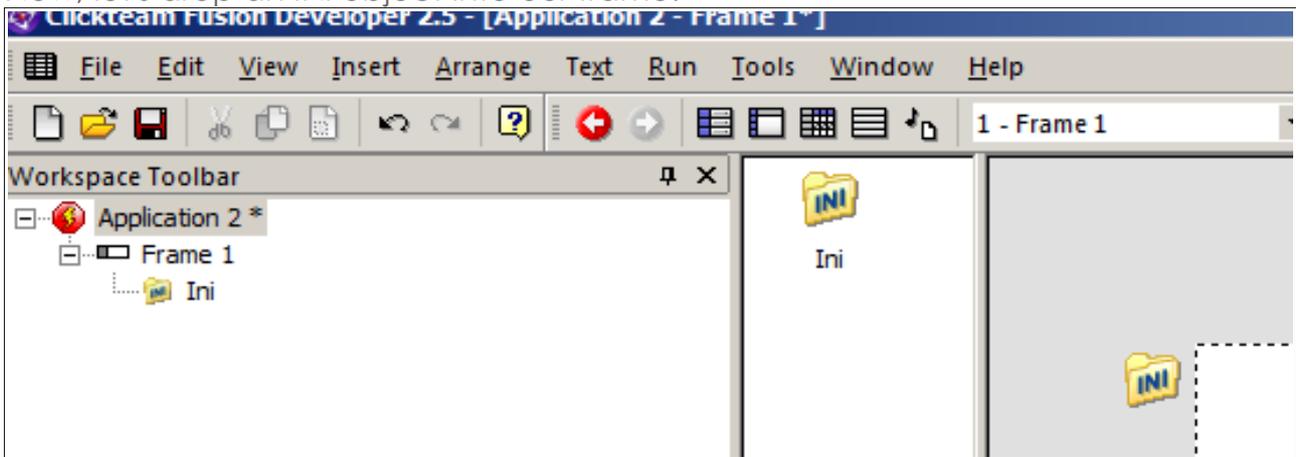
Once you have an INI object in there, click on it and uncheck the property 'Create INI file in Application Folder'. We do not need this property as we're not creating an INI, we will already have included the INI file in our application, we just need to load it from binary at runtime.

Now, drop a counter into the frame.
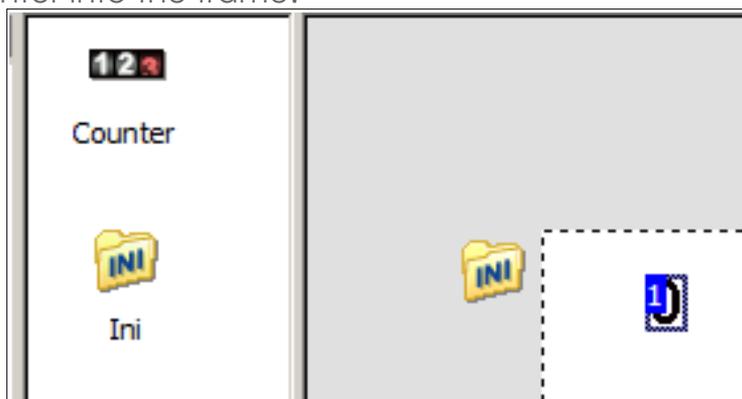
Now our frame is setup and ready for us to use the INI file. Proceed to the next sections.

4. CALLING BINARY DATA IN WINDOWS

So, here is the fully fledged method of calling this INI file from Windows.

Please Note, whilst testing your application (via run frame or run application) the loading of this file will come directly from c:\test.ini as it will not be included into binary until you come to do your final build.

One other note to make is, we do not need to set the filename in the property of the INI. We will do that via events. Let's jump into the event editor.

First of all, we need to EXTRACT the binary file we want to use. The reason for extracting it is because, unless we need to use the file, it's pointless extracting files to the users temporary directory. So let's go ahead and extract it.
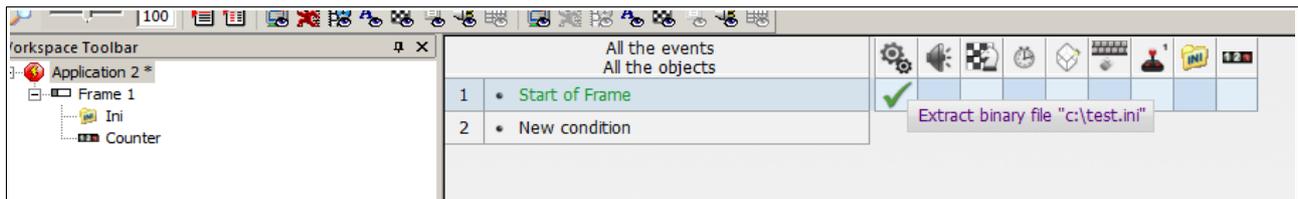
So, in the event editor create a new condition, Start of Frame. We will extract the INI at the Start of the Frame as we know we're going to be using it immediately. When you've created the Start of Frame condition, right-click under the Special Object and select **Binary Files > Extract Binary File**. In the prompt type "c:\test.ini" and click OK.

This will now extract the binary file from your application into a temporary folder on the users computer. Now, with the way Windows and Fusion works, when it extracts the temporary binary file called test.ini, it won't be extracted AS test.ini. When it's extracted to the temporary folder it will usually have an obscure filename but no fear, as we can obtain this filename via a simple expression.

To keep our MFA looking clean and simple for this guide, let's create another Start of Frame condition, separate from the first one. In this new condition, hover over to INI, right-click and click on '**Set Current File**'
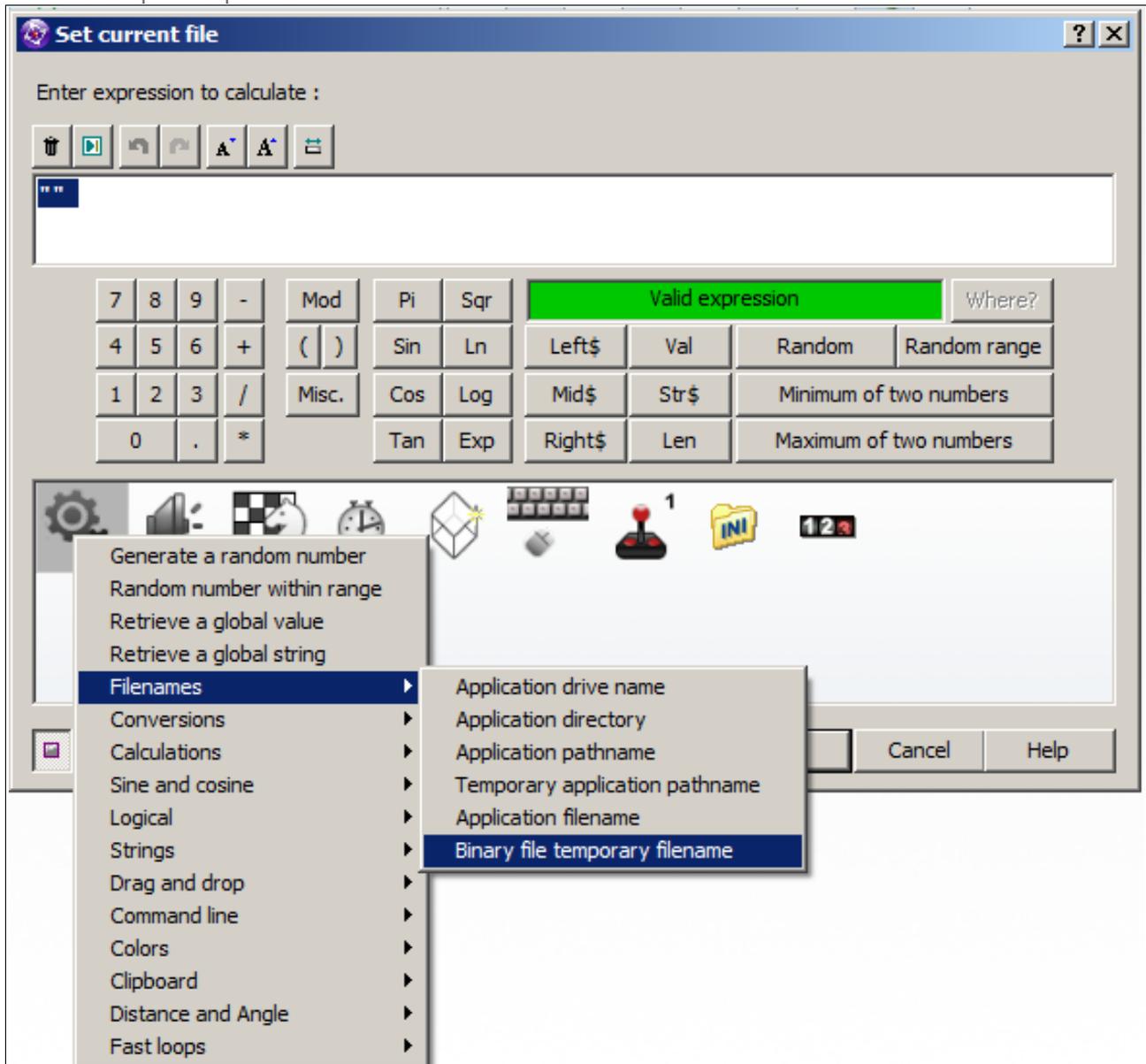

For more information, tutorials, examples and walk-thrus visit the website.

4. CALLING BINARY DATA IN WINDOWS

When prompted for the filename, we cannot just type "c:\test.ini" as this will not be the 'actual' windows filename. We can retrieve what the temporary filename is with a simple expression.
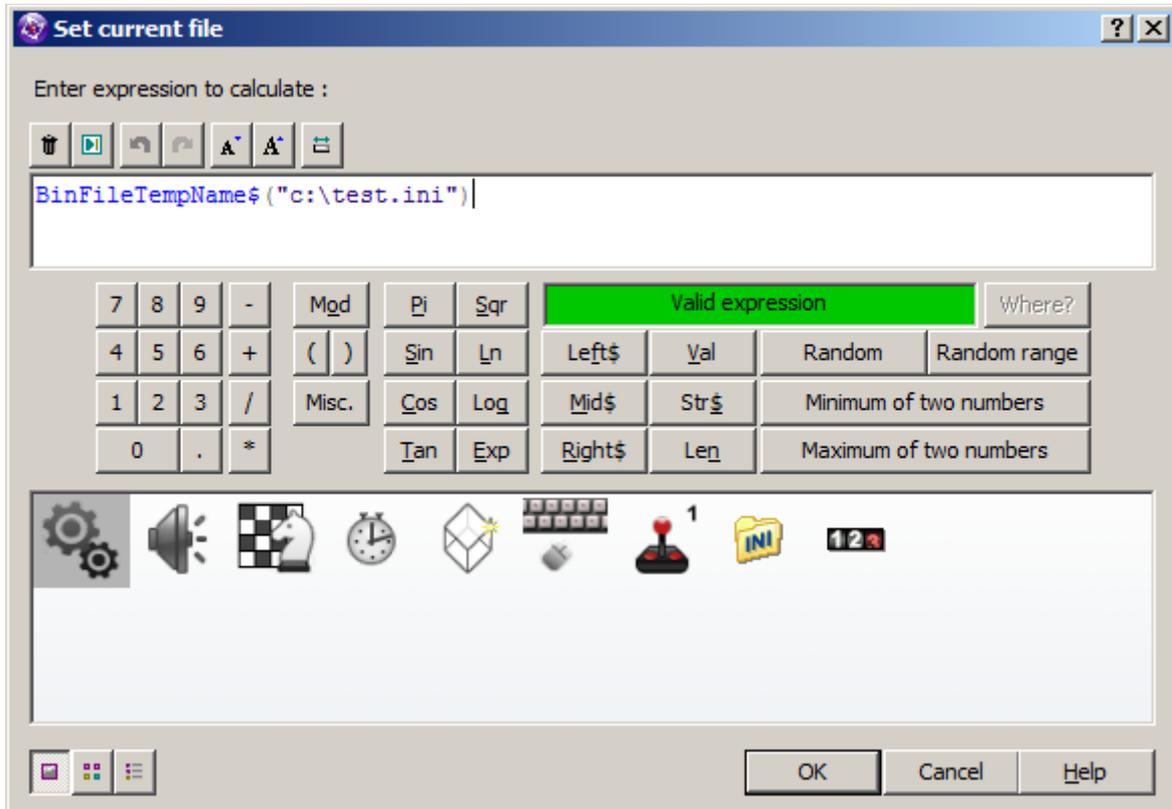

Once you have click on Filenames > Binary file temporary filename, it will then ask you which file you want to reference. In this instance, we want to reference "c:\test.ini" so we need to type that in the expression editor like so...

For more information, tutorials, examples and walk-thrus visit the website.

4. CALLING BINARY DATA IN WINDOWS

Your expression editor should now look like this.

Now, let's put this to the test. Let's create another Start of Frame event and we will load the value in that INI we created which was 101, into the counter. So, under the new Start of Frame condition, hover over to the counter, right-click and **set counter**.
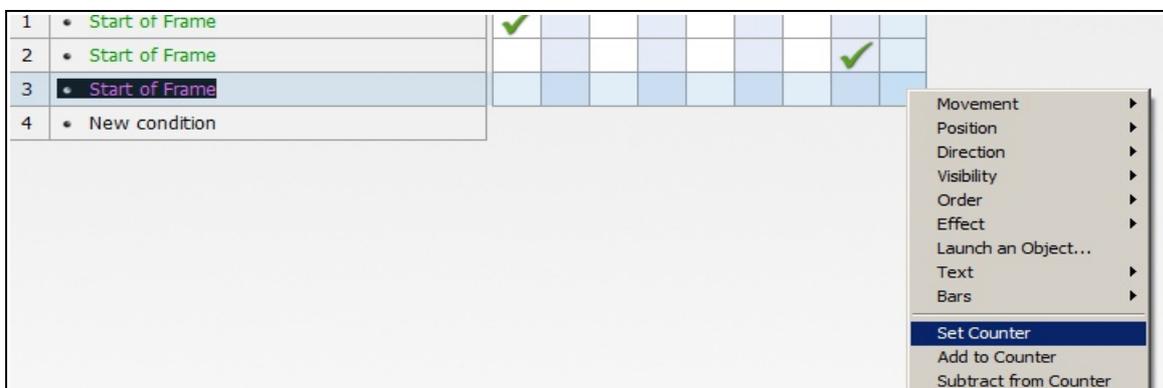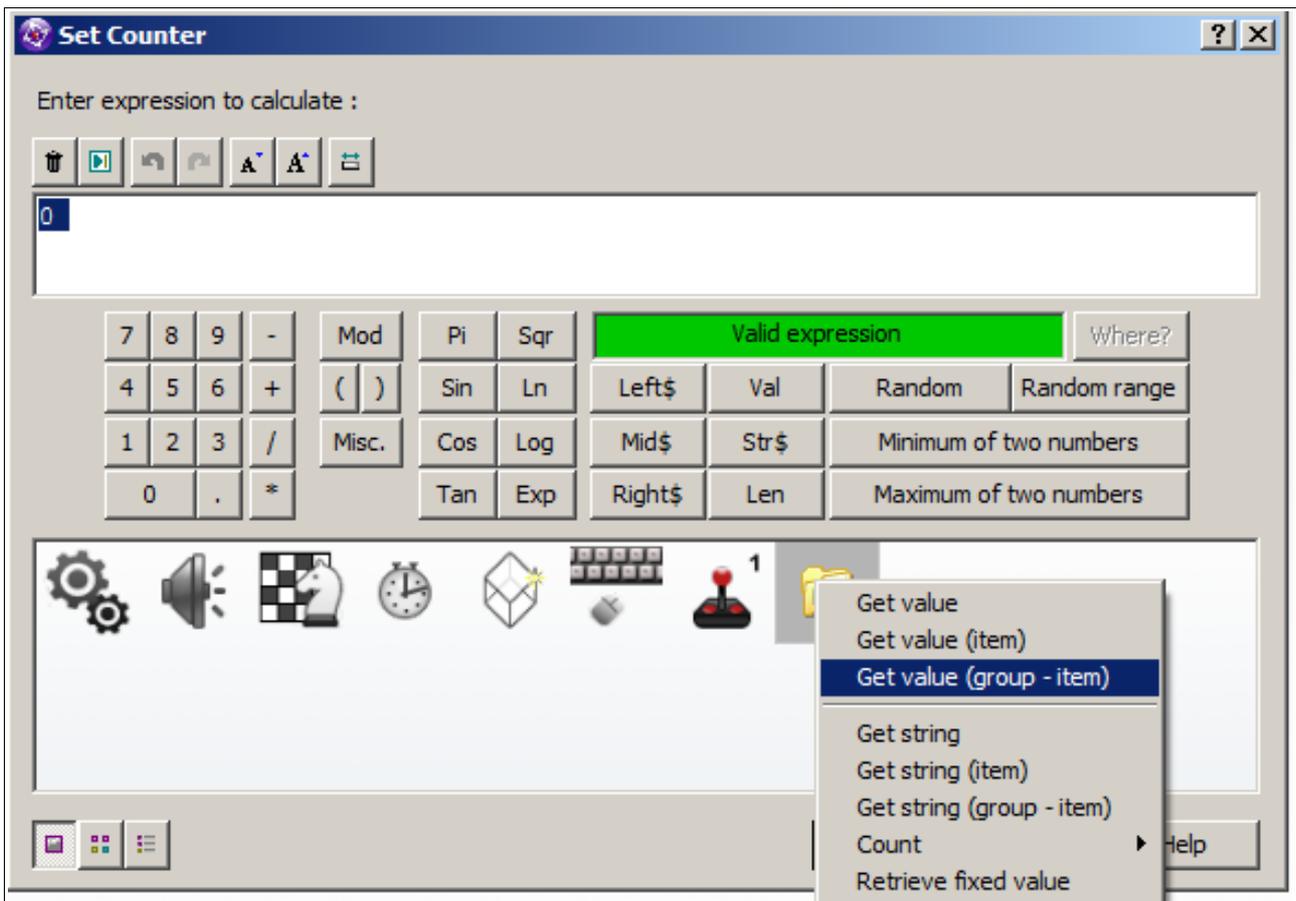

For more information, tutorials, examples and walk-thrus visit the website.

4. CALLING BINARY DATA IN WINDOWS

When the expression editor pops up, we need to load the value from the INI to the counter, so go ahead in the expression editor, right-click on the INI and select **'Get Value (group-item)'**
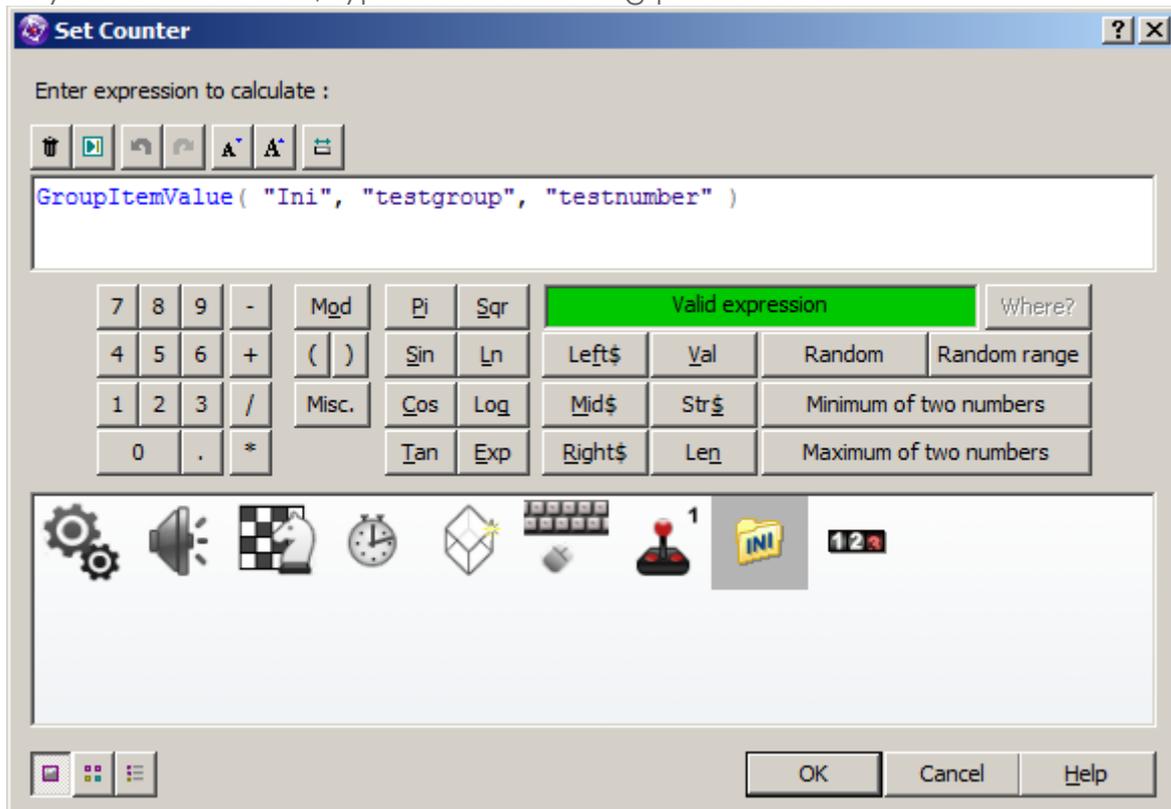
4. CALLING BINARY DATA IN WINDOWS

Once you've done that, type in the following parameters:

As you can see, we are asking the expression editor to retrieve the current value from the INI file, under the group 'testgroup' and under the item 'testnumber'. Now, that should be almost complete.

Hit F8 and run the application. The application should display the counter showing 101. This is successful. Now, because we haven't BUILT our final application, it will just use the actual c:\test.ini file for testing, once you compile and do your final EXE build, it will include the test.ini file as binary and your application will now be able to reference that file.

You have successfully been able to now include a binary file, extract a binary file and read a binary file. Finally, you will need to 'release' the binary file. If you don't release the binary file before your application closes, then it will stay in the temporary folder, taking up unwanted space on the users hard drive.
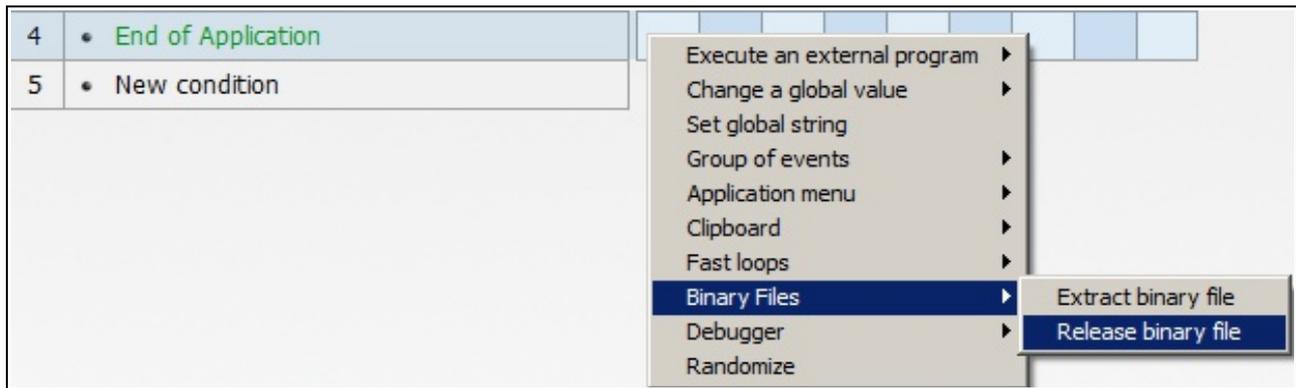
4. CALLING BINARY DATA IN WINDOWS

A quick way to release any binary files is to do it on the End of Application event. So let's create a new condition: End of Application.

Hover under the special object, select **Binary Files > Release Binary File**.
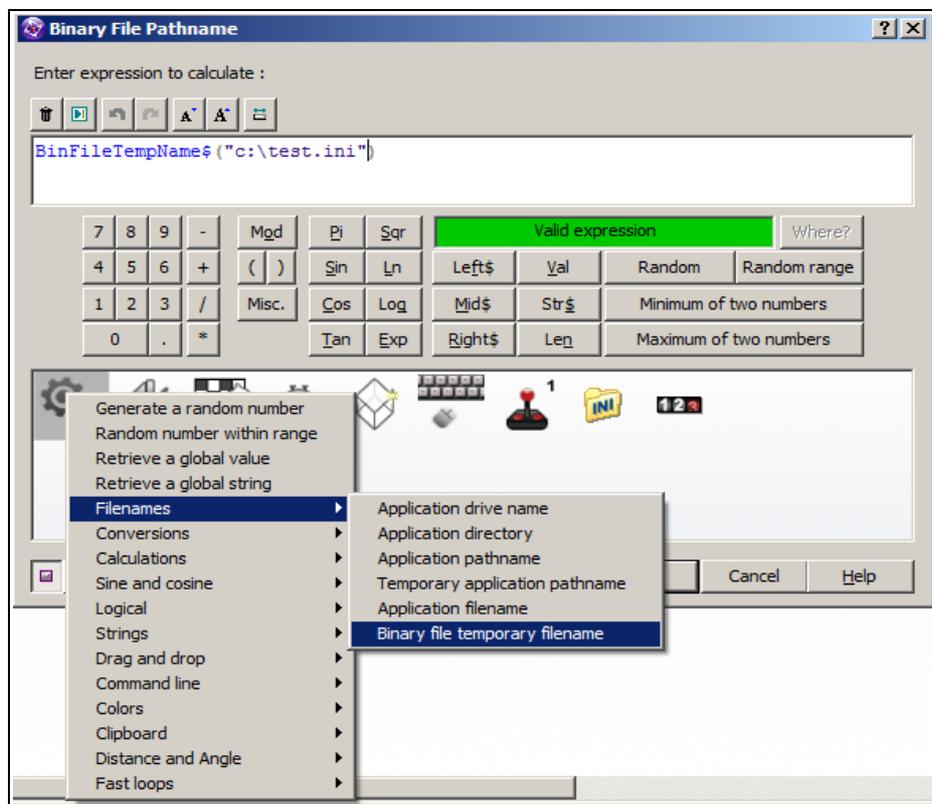
For more information, tutorials, examples and walk-thrus visit the website.

4. CALLING BINARY DATA IN WINDOWS

Your expression editor should look like the screenshot above. When releasing the file, we need to tell Fusion which file to release, we want to release c:\test.ini but remember, when stored in the users temporary files, it has an obscure filename so we retrieve the 'actual' filename via a simple expression of

BinFileTempName$("c:\test.ini")

That's practically all there is to know about binary data and extraction with Clickteam Fusion.

Next we will cover how to use Binary Data in the exporters. It literally is, 100x quicker and easier than the Windows method.

5. CALLING BINARY DATA IN THE EXPORTERS

If you re-call the above method for Windows, the exporters is very simple. You do all of the above except extract and release.

So, if you were compiling a game for Android, you would call the file exactly how it appears under the Data Elements button. For example, if it was c:\test.ini like so:
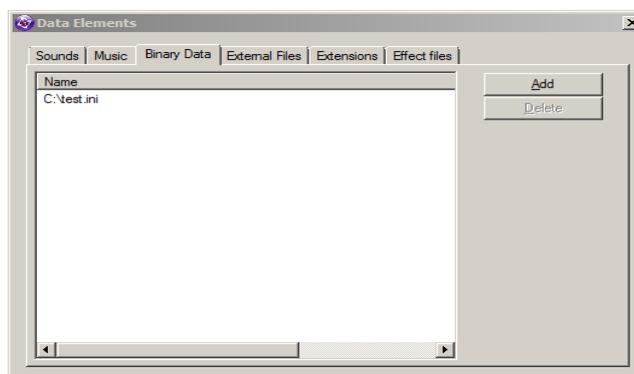
Then in your event where you want to LOAD the ini file, you would just hover under the INI, right-click and select **'Set Current File'** to "c:\test.ini".
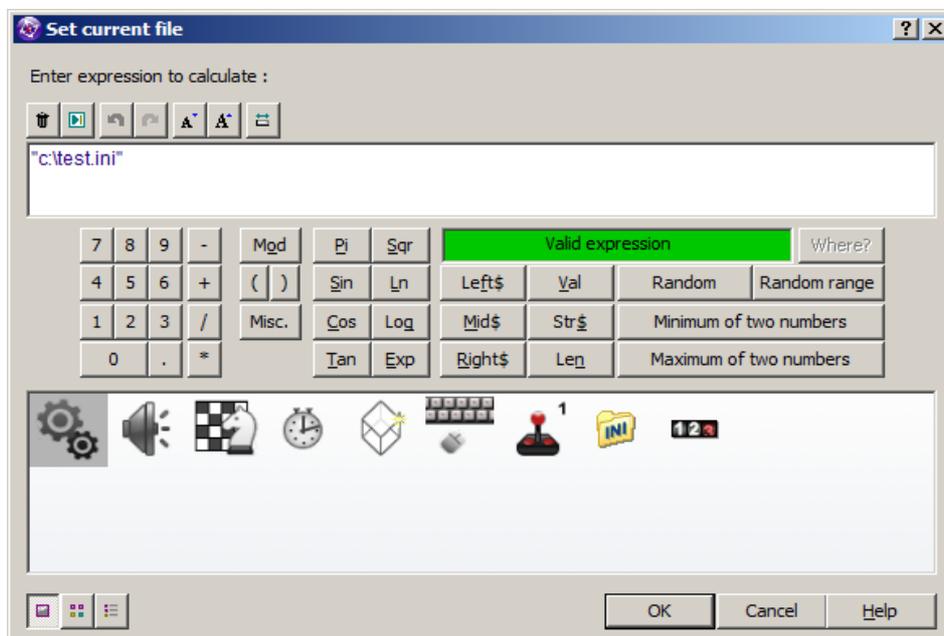
It REALLY is as simple as that. This method will work across all exporters including XNA, Android, iOS, Flash and HTML5 when using Binary Data under Data Elements.

For more information, tutorials, examples and walk-thrus visit the website.

6. CONCLUSION

So, in this guide we have practically covered how to use Binary Data in Fusion with your applications and games across all platforms.

Binary Data can be a very powerful feature to use. I strongly suggest you take the time to trial and error some other files such as images, sounds and much more. Don't forget, until you BUILD your final application, it will always reference the local files. Only once you have BUILT your final application, Fusion will compile these files as binary data in your executable.

I hope you've enjoyed this guide and I hope it has equally been just as much use to you.

For more information, tutorials, examples and walk-thrus visit the website.

Website: http://www.Clickteam.com – **Community:** http://community.clickteam.com