

Part 1: Introduction to Ini files by Whiskers

What are ini files?

While there are numerous sites offering help with ini files, I wanted to share with the novice how they are used in this application, Brain Blaster. Perhaps this info will clear up any confusion the reader may have previously had. To begin, let me encourage the beginner that ini files are easy to work with. They're nothing more than a simple text file that are used to store values (i.e. numbers) or strings (i.e. text). The file can easily be read in note pad or a similar program. The files allow the user to store info (i.e. a player's name, score, etc.) that can be accessed later during the game or saved for future use after the game play has finished. I will keep this explanation overly simplified.

Saving and loading strings

Let's start in a new frame by inserting the ini object. Do this by right clicking on the frame and then left clicking on *Insert Object*. In the *Create New Object* menu you will find the ini object under the listing, *Storage*. Click on it, then click OK, followed by clicking on the frame. Double click on the ini object to open a small window. In this window you will type where you want your ini file located. Just to keep it simple, we will use the root directory. So, type this: c:\game.ini in the window and press enter. Later we will change this location to something more practical.

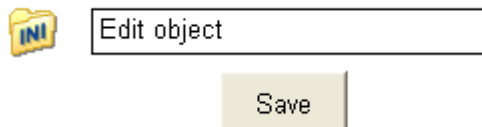
Assign a "group" name and "items" to your ini file

The information that you wish to store in your file is assigned to an "item". All the items are contained within a group. It would look like this:

```
[Player 1]
name=Bobby
score= 1024
```

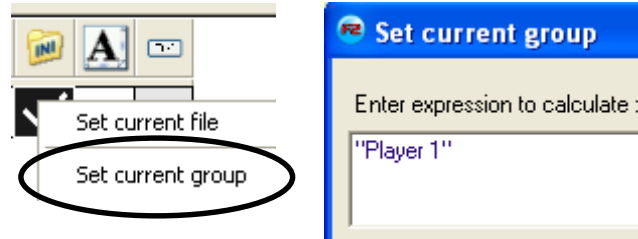
The name of the group is "Player 1". The first item is called "name". The second item is called "score". If this were a real example, it would mean this: Bobby decides to play the game. The game ask that Bobby enter his name. The item "name" stores this entry. When exiting the game Bobby's score is saved in the game under the item "score". Later, when Bobby plays the game again he clicks on Player 1 from the game menu and Bobby's name and score are displayed.

We'll now set up some simple actions to see how to save and access information from the ini file. In your frame, insert an Edit Box. You will find this under the category *Interface* in the *Create New Object* menu. We also want to put a button in the frame. This button object is also found under the category *Interface*. Double click on the button and change the name of it to "Save".



What we wish to accomplish is this: The user will type his/her name into the box, then click on the button to save the name to the ini file. We will accomplish this in the Event editor.

In the Event editor, make an event called Start of Frame. Right click under the ini icon and left click on *Set current group*. Type in "Player1" (with quotes) in the Expression editor.

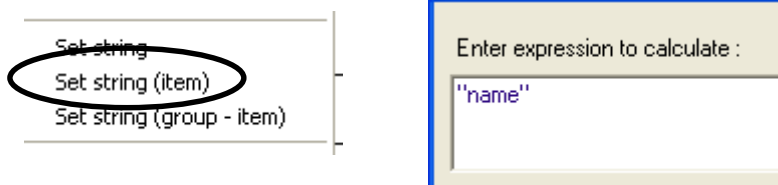


The group, Player 1 will now be the group to which all entered information is saved to when this frame is opened. Our desire is to save the player's name to this group. The action will follow this logic. When the button is clicked the text that's in the edit box will be saved in the ini file. Let's do this by making a new event:

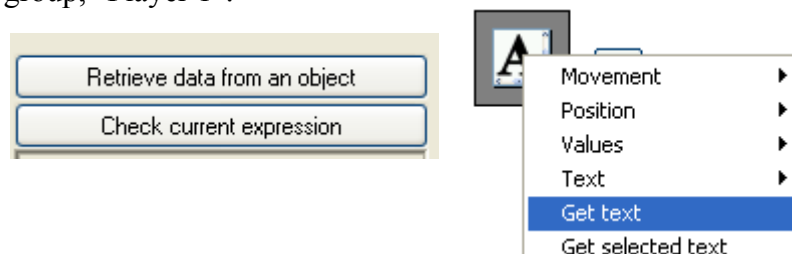
Button is clicked (right click on New Condition > Button > Button clicked?)



What do we want to happen when the button is clicked? We want to take the text (string) entered into the edit box and store it into the ini file. We will do this now. Right click under the ini icon on this same line as Button clicked, and click on *Set string (item)*. In the Expression editor, type a name of your choice. We will simply use the word, "name" in our example.



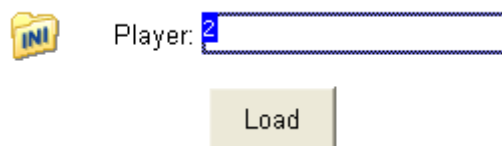
When you click OK in the Expression editor, the editor refreshes itself with another inquiry. It's asking what do you want to save with the item called "name". What we want to save is this: whatever the user has typed into the edit box. This is what we tell the Expression Editor. So, do this: Click on *Retrieve data from an object*, followed by right clicking on the Edit box and selecting, *Get text*. Click OK to finish. We told the Expression editor to get the text from the edit box and save it in the ini file, identified with the item called "name". All this is saved in the group, "Player 1".



Give it a try. Run your frame, enter your name in the edit box and click on the button. Now take a look in your ini file by going to your C drive and opening the file called "game". You should see this:

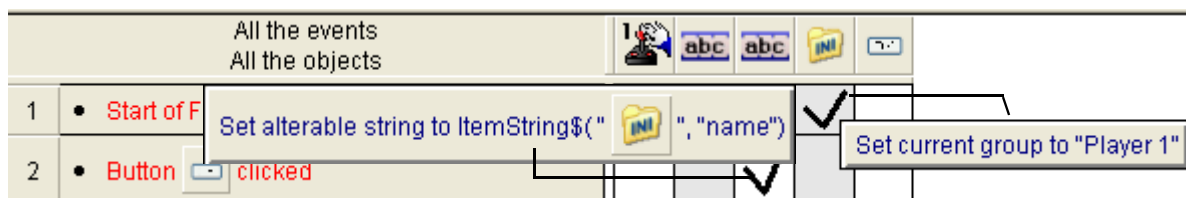
```
[Player 1]
name = Your name
```

Now, we've got to retrieve this saved info if we're going to make it useful. What we will do, is this: make a second frame and insert a string (Create new object > text > string), which reads, "Player:" Reduce the size of the text box to only what is needed to hold the text. Next, put another empty string box just to the right of it. It's within this empty box we will have the player's name appear. Beneath this, put a button and change the name to "Load". The whole thing looks like this:

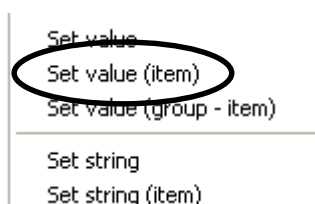


Next, insert another ini object onto the frame. As before, double click on this icon to open the text box. Type within it C:\game

Let's move to the Event editor to make this whole thing work. As we did in frame 1, we will make our first event "Start of Frame". Again, like in frame 1, right click under the ini object and click on *Set current group*. As before, make this group "Player 1". As in frame 1, make a second event for clicking the button (Button clicked?). The action that corresponds with this event (the action that will put the player's name in the text box) is done this way: Right click under the string 2 object and click on *Change alterable string*. In the Expression editor, click on *Retrieve data from an object*. Right click on the ini object, followed by clicking on *Get string (item)*. In the Expression editor that appears we want to change >item name< (it may be highlighted) to the actual name of the item, which in this case is "name". So, delete >item name< and type in "name" (use the quotes). The end result should look like the figure below. Test it out by running the first frame, followed by running the second.



The preceding tutorial dealt with saving and loading a string. The same procedure would apply to working with values. Just make sure you select the correct options, as illustrated:



Part 2: Advanced Ini Saving and Loading

Let's suppose you've designed a game where you desire more than one player to log in and save their name and score. No problem. There are a number of ways to accomplish this. This is just one of them. The end result of the interface (when ran) will look like the image below:



The following objects are put onto frame 1: **(a)** A text string that reads "Player"; **(b)** a text string that reads "Type in your name and press Enter"; **(c)** a text string that reads "Score" **(d)** two counters (starting at 0); **(e)** a button; **(f)** an edit box; **(g)** the ini object set to save the info somewhere on your hard drive.

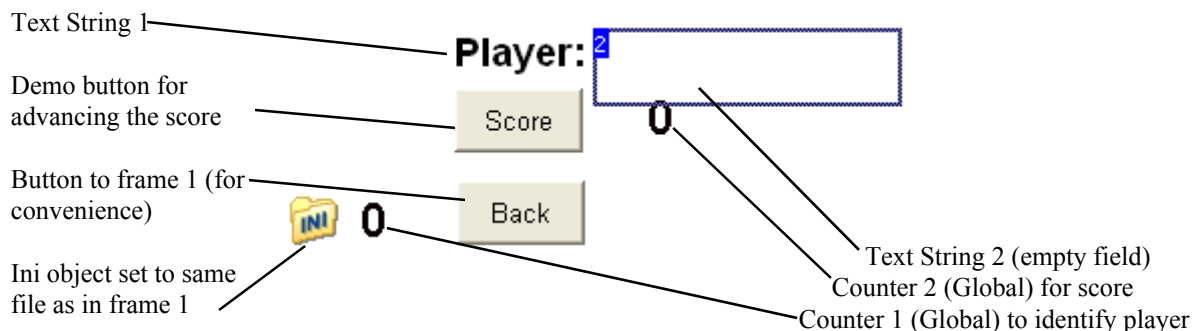
The way it will work, is like this: he or she clicks on the button and a blank entry field appears in the edit box. The user types in his/her name and pushes the enter key, as requested. Frame 2 now appears with the player's name on the screen. Should someone else wish to log in, they would click the button until a blank field appears (bypassing any other names previously entered) and put in their name. As the button is clicked, the player number advances by one number. Any player can then return to this log in screen, find their name and see their current score, push enter and resume their game playing. While you can design the log in screen to handle any number of players, this tutorial will cover five entries. So as not to tire the reader with a lot of reading, I will simply give the instructions in a step format. *Details for any steps worthy of further explanation are given later.*

Step by step instructions:

1. Make two frames for the application. Frame 1 will be the log in frame. Frame 2 will be the game playing frame.
2. In frame 1 insert the objects mentioned previously: (a) through (g). Arrange them according to the image above.

3. On frame 2, as illustrated below, insert a text string that reads, "Player". Just to the right of it, put another text string onto the screen. Delete the default word, "Text", leaving the field empty.
4. Put counter #1 on frame 2, out of view of the playing area. Make this counter global (see notes at conclusion for making a counter global) with counter #1 on frame 1.
5. Place counter #2 on frame 2. Make this counter global with counter #2 on frame 1.
6. Place a button, labeled "Score" onto frame 2. Another button (optional) is placed upon frame 2 to allow you to click back to frame 1.
7. Place the ini object, set to the same file location as the ini object of frame 1, on frame 2
8. Arrange the objects, as illustrate later below.

Frame 2



Making it All Work in the Event Editor

Go to the Event Editor of frame 1 and set these events (see the illustration later on, below):

Event 1 This event has the Edit Box displaying the message: "Click Button - Enter Your Name". It also resets both counters to 0.

Start of Frame

Under the Edit Box : Right click > Editing > Set Text. Between the quotes in the Expression Editor type, "Click Button - Enter Your Name". Click OK.

Under the Counters: Right click > Set counter to 0. Click OK.

"Player 1" > OK > type this (with quotes) "Name" > OK > click on *Retrieve data from an object* > right click on Edit Box > Get text > OK.

Event 2 This cause the player number to advance by one each time the button is clicked.

Button Clicked (Right click on New condition > Button > Button Clicked?)

Under Counter 1: Right click > Add to counter > 1. Click OK.

Event 3 When counter (1) is equal to 1 then the name of player 1 in the Edit box is saved to the ini file.

Counter 1 = 1 (Right click on New condition > Counter > Compare the counter to a value > 1. Click OK.

Events 4 - 7 These events are the same as event 3, except the group name is changed to either Player 2, Player 3, Player 4, or Player 5 when requested in the Expression Editor.

Under the Ini icon: Right click > Set string (group - item) > type this (with quotes)

Events 8 -12 These events load the saved names from the ini file into the Edit Box. They also change counter 2 to the saved score in the ini file. These events are identical, with the exception of each group name. That is, the group name for event 8 is Player 1; event 9 = Player 2; event 10 = Player 3; event 11 = Player 4; event 12 = Player 5.

Button Clicked

Counter 1 = 1 (or = 2, 3, 4, or 5 in the other events)

Under the Edit Box: Right click > Editing > Set text > Retrieve data from an object >

Right click on the Ini object > Get string (group - item) > Replace > *Group name* < with "Player 1" > Replace > *Item name* < with "name" > OK.

Under Counter 2: Rick click > Set counter > Retrieve data from an object > Right click on the Ini object > Get value (group - item) > Replace > *Group name* < with "Player 1" > Replace > *Item name* < with "score" > OK.

Event 13 When the counter advances to 6, both counters are reset to 0 because we are only allowing five players to log in.

Event 14 When the counter equals 0 then the message reappears back in the Edit Box.

Event 15 Advances to frame 2.

Event 16 Resets the score (counter) to 0.

Go to the Event Editor of **Frame 2** and make the following entries:

Event 1

This event load the player's name into the empty text field.

Counter 1 = 1

Under String 2: Right click > Change alterable string > Click on *Retrieve data from an object* > Right click on Ini > Click on *Get String (group - name)* > Replace > *Group name* < with "Player 1" > Replace > *Item name* < with "name" > OK.

Events 2 - 5

All four events are identical with event 1, with the exception that the group name is different for each event. That is, the group name in event 1 is Player 1; event 2 is Player 2; event 3 is Player 3; event 4 is Player 4; event 5 is Player 5.

In addition, the font size for the player's name has been set to 14 pt. This is easily done, as follows: Right click under string 2 > Text > Set font size > Enter 14 > OK > Select option 1 to accept > OK.

Events 6 - 10

In events 6 through 10, the player clicks on the Back button, which in turns, saves the counter 2 value (the score) to the Ini file. In addition, the player is sent back to frame 1. The counter is used to identify the player with the action.

Event 6

(Back) Button clicked

Counter 1 = 1

Under the Ini object: Right click > Set value (group - item) > Enter (with quotes) "Player 1" > OK > Enter (with quotes) "Score" > OK > Click on *Retrieve data from an object* > Right click on Counter 2 > Click on *Current value* > OK.

Events 7 - 10





























(Back) Button clicked

Counter 1 = 2 (or Counter 1 = 3, 4, or 5) On each of the events, the value of counter 1 is set differently, with each action equaling a different group name. That is, if the counter is made to equal 2, then the group name "Player 2" is entered in the Expression Editor, etc.


















Event 11

When the button labeled "score" is clicked, the value of counter 2 is increased by 3.

Frame 1

All the events All the objects							
1	• Start of Frame		✓		✓	✓	Set Counter to 0
2	• Button  clicked				✓		Add 1 to Counter
3	•  = 1	✓					Set text "Click Button -Enter Your Name"
4	•  = 2	✓					• Set string Edittext\$("  ") to item "name" in group "Player1" • Set value value("  ") to item "score" in group "player1"
5	•  = 3	✓					
6	•  = 4	✓					
7	•  = 5	✓					
8	• Button  clicked •  = 1	✓				✓	
9	• Button  clicked •  = 2	✓				✓	
10	• Button  clicked •  = 3	✓				✓	Set Counter to GroupItemValue("  ", "player1", "score")
11	• Button  clicked •  = 4	✓				✓	
12	• Button  clicked •  = 5	✓				✓	
13	• Button  clicked •  = 6				✓	✓	Set Counter to 0
14	• Button  clicked •  = 0		✓				Set text "Click Again, No Entry Here"
15	• Upon pressing "Enter"	✓					




Frame 2

All the events All the objects											
1	<ul style="list-style-type: none"> •  = 1 	<ul style="list-style-type: none"> • Set alterable string to GroupItemString\$("INI", "Player1", "name") • Set font size to 14, border 1 					✓				
2	<ul style="list-style-type: none"> •  = 2 						✓				
3	<ul style="list-style-type: none"> •  = 3 						✓				
4	<ul style="list-style-type: none"> •  = 4 						✓				
5	<ul style="list-style-type: none"> •  = 5 						✓				
<ul style="list-style-type: none"> • Button  clicked •  = 1 		Back Button					✓		✓		
<ul style="list-style-type: none"> • Button  clicked •  = 2 							✓		✓		
<ul style="list-style-type: none"> • Button  clicked •  = 3 		Counter 1 is used in events 6 - 10					✓		✓		
<ul style="list-style-type: none"> • Button  clicked •  = 4 							✓		✓		
<ul style="list-style-type: none"> • Button  clicked •  = 5 							✓		✓		
11	<ul style="list-style-type: none"> • Button  clicked 									<div>Add 3 to Counter</div>	✓


Part 3: Using the Ini File in the Brain Blaster game Frame 1

- Set current group to "player1"
- Set current file to Appdrive\$+Appdir\$+"totals.ini"

Some events hidden
All the objects

1	• Start of Frame
2	• User clicks with left button on 
3	• User clicks with left button on 
4	• User clicks with left button on 

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Set text GroupItemString\$("  ", "player1", "name1")

In frame one of the game, we have these events related to the ini file. On the Start of Frame event, we have the group set to player 1. The ini file has been set to save the data within the game itself. This is done by right clicking under the ini object and then clicking on *Set current file*. In the Expression Editor type this: Appdrive\$+Appdir\$+"totals.ini" You can name the ini file anything you like. I've named it *totals*. Make sure to include the quote marks. Saving the data within the game itself is the preferred way of working with the ini file.

The next event has this action: When the user clicks on the text string of Player 1, the player's name is loaded into the text box. Of course, if there is no saved name in the ini file, the box will remain empty ready for the player to type in a name. Event 3 follows the same logic: When the player clicks on the text string of Player 2, the action under the edit box is set to "player2", "name2". Event 4 is set for clicking on the text string of Player 4, setting the action to "player3", "name3".

Some events hidden All the objects				
5	<ul style="list-style-type: none"> = 1 Upon pressing "Enter" 		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	<ul style="list-style-type: none"> = 2 Upon pressing "Enter" 		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	<ul style="list-style-type: none"> = 3 Upon pressing "Enter" 		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Set string Edittext\$(" ") to item "name1" in group "player1"
--

Event 5 works like this: The player has the option of clicking on either player 1, 2 or 3 when logging in. If player 1 is clicked, the counter is set to 1. Pushing the enter key saves the name entered, as illustrated. Event 6 follows the save logic, setting the counter to 2 and saving the data to "name2 in group "player2". Event 7 saves to "name3" in "player3". This counter is a global counter, given the name "player".

Frame 2

Some events hidden All the objects									
1	• Start of Frame	✓	✓					✓	✓
2	• Button clicked + = 1			✓	✓	✓	✓	✓	
3	• Button clicked + = 2			✓	✓	✓	✓	✓	
4	• Button clicked + = 3			✓	✓	✓	✓	✓	
5	• = 1			✓	✓	✓	✓	✓	
6	• = 2			✓	✓	✓	✓	✓	
7	• = 3			✓	✓	✓	✓	✓	

<ul style="list-style-type: none"> Set value 0 to item "scorebible" in group "player1" Set value 0 to item "scoreanimals" in group "player1" Set value 0 to item "scorehistory" in group "player1" Set value 0 to item "scoremania" in group "player1" 	Set current group to "player1"
--	--------------------------------

Events 2, 3, 4 allow the user to reset the score of each category back to 0 when the button (labeled reset) is clicked on. Event 2 pertains to player 1; event 3 pertains to player 2; event 4 pertains to player 3.

Events 4, 6, 7 deal with the global counter. Depending on what the counter equals, decides what the current group is set to.





Some events hidden All the objects					
5	• = 1	✓	✓	✓	✓
6	• = 2	✓	✓	✓	✓
7	• = 3	✓	✓	✓	✓

Set Counter to GroupItemValue(" ", "player1", "scoreanimals")	
Set Counter to GroupItemValue(" ", "player1", "scorebible")	
Set Counter to GroupItemValue(" ", "player1", "scorehistory")	
Set Counter to GroupItemValue(" ", "player1", "scoremania")	

Event 5 says, when the counter (a global counter named player) is equal to 1, then set these counters to the score saved in the ini file. Event 6 says the same, except it is for player2. Event 7 says the same, except it is for player3.




The Games



All four categories process the saving of data to the ini file the same. Here is one of the categories. As in all frames, the ini file is set at the Start of Frame event. Event 2, below, is an action that instructs the final score to be saved to the ini file. That is, set the current value of the counter called "biblescore". Again, the global counter named "player" determines who is playing the game (either player 1, 2 or 3) and where to set the current group and the item to save the score to.

Some events hidden All the objects			
1	• Start of Frame	✓	Set current file to Appdrive\$+Appdir\$+"totals.ini"
2	•  has reached the end of its path	✓	Set item value value(" 0 ") biblescore
3	•  = 1 player	✓	• Set current group to "player1" • Set current item to "scorebible"
4	•  = 2	✓	
5	•  = 3	✓	

The Score Totals

This is the frame that displays the accumulated scores. Again, the counter named "player" is used to determine who the player is. The score counters are set according to what has been written and stored in the ini file. The action of the string object inserts the player's name onto the frame.

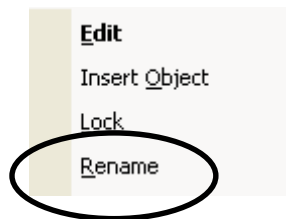
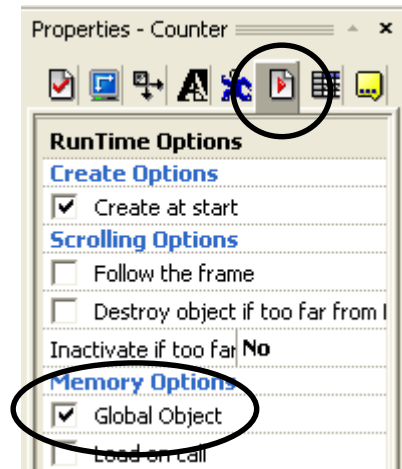
Some events hidden All the objects		animalscore	historyscore	biblescore	maniascore	
1	• Start of Frame	0	0	0	0	✓
2	•  = 1	✓	✓	✓	✓	✓
3	•  = 2	✓	✓	✓	✓	✓
4	•  = 3	✓	✓	✓	✓	✓

Set Counter to ItemValue("  ", "scoreanimals")	<ul style="list-style-type: none"> Set alterable string to ItemString\$("  ", "name1") Set font size to 12, border 1 Set bold 1
---	---

Notes

In order for the game to save to and access from the ini file, the counter for each of the three players has to be made global. Making the counter global simply means that the value of the counter is carried over from frame to frame. This is a simple process to do, as I'll describe here.

1. Put a counter onto a frame.
2. In the Properties menu, click on the Global Object box located under the tab shown circled at the top of the image.
3. Right click on the counter to bring up the list of options and click on Rename. Give the counter a descriptive name.
4. Make a second frame and repeat the same three steps above.
5. To see it work, add an action in the Event editor that increases the counter by a few numbers. Put another action that allows you to toggle between the two frames to view each counter.



Hope you found this tutorial helpful. Whiskers

Email: stthread512@cox.net

All Rights Reserved Copyrighted 2008 S. D. Threadgill