

Mokhtar M. Khorshid's Random Multipool Extension

About The Random Multipool

The Random Multipool object is the evolution of the Random pool in the MMF2 bonus pack (which was written by Michael Saad and Mokhtar M. Khorshid). The purpose of the extension object is to simplify the management of non-repeating random numbers. This purpose is realized via two main features. The first is its ability to generate predictable sequences of random numbers if given the same seed. This is important because it simplifies debugging by allowing the developers to regenerate the same sequence of numbers the player got when some bug occurred; they only need to save the seed that was used. The second feature, which is the main improvement over the original Random Pool, is the ability to maintain multiple pools simultaneously. Pools are identified using a string Pool ID which can be anything you want. Almost all the functions of the extension will require a Pool ID to specify which pool they operate on. Each pool will hold a different set of numbers that you can randomly pull numbers from.

This extension was inspired by Nico and was written by Mokhtar M. Khorshid but uses some code from the original Random Pool object which was written partially by Michael Saad. Report any bugs or request support for this extension at <http://www.thunder-power.com/support>.

Conditions

Pool Is Empty (Pool ID)

Returns true if the specified pool is empty.

On Number Generated (Pool ID)

This is a true triggered event that is triggered once a number has been generated successfully.

On Number Iterated (Pool ID)

This triggered event is fired once for each number during a loop. It should be used with the Get Iterated Number expression.

Actions

Fill Pool (Pool ID)

This action fills the specified pool with numbers. The numbers will span the complete range specified using the Set Limits action.

Set Pool Limits (Pool ID, Min, Max)

This action sets the lower and upper limits for the values in the pool. This will also restrict numbers you add yourself. However, if you change the limits while it is not empty, the numbers already in the pool will remain unaffected. You will usually want to do this before using a pool. The range between Min and Max cannot exceed 1000000 but there are no restrictions on the actual values you use for Min or Max.

Set Manual Seed (Pool ID, Seed)

This action sets the seed to a specific value. Using this will make your generated sequences predictable (i.e. you will get the same sequence each time). If you need truly random numbers either set this seed to a random number from MMF or use the [Shuffle](#) action.

Load Pool from String (Pool ID, String)

This action loads the values in the string into the pool, in addition to whatever was there in the pool. The string should have only numbers separated by whitespaces like "3 5 6 2".

Load Pool from File as List (Pool ID, Filename)

This action will load and add a list of numbers from a file into the pool. Each number should be on a separate line, but the numbers can only be separated by white spaces.

Save Pool to File as List (Pool ID, Filename)

This action will write the contents of the pool to the specified file one number per line.

Load Pool from File as String (Pool ID, Filename)

This action will load numbers from the specified file as a string of numbers delimited by strings that do **not** contain white spaces in the middle. So the file can have "1,2,3" or "1 p 2 d 3" but not "1 p p 2 d d 3."

Save Pool to File as String (Pool ID, Filename, Delimiter)

This action will save the pool to a file as a string of numbers delimited by the specified delimiter. The delimiter can be anything you like, but if it contains spaces in the middle you will not be able to load it using the [Load Pool from File as String](#) action.

Add a Number (Pool ID, Number)

This action will add the specified number to the specified pool. If the number is not within the pool's limits, then the action is ignored. You can add a number even if it already exists in the pool.

Remove All Occurrences of Number (Pool ID, Number)

This action will remove all the occurrences of the specified number from the specified pool. Note that since the pool allows duplicates, this will remove all such duplicates of the number.

Generate a Number (Pool ID)

This is probably the most important action; it pulls out a number randomly from the pool. If the action is successful the [On Number Generated](#) event will be triggered. This will fail if the pool is empty.

Iterate over All Numbers in Pool (Pool ID)

This action will loop over all the numbers in the pool. For each number the [On Number Iterated](#) event will be triggered once, just like fast loops except that it is identified by the pool ID.

Iterate over All Utilized Numbers in Pool (Pool ID)

Similar to [Iterate over All Numbers in Pool](#), this action will iterate over all the utilized numbers of the specified pool. For each number the [On Number Iterated](#) event will be triggered once, just like fast loops except that it is identified by the pool ID.

Shuffle/Randomize Seed (Pool ID)

This action will shuffle the pool, or equivalently randomize the seed. You will need to use this action if you do not want to generate predictable sequences. By default, the seed will not be randomized.

Sort (Pool ID, Order, What to Sort)

Each pool has an internal ordering of the numbers it holds. When pulling numbers out of the pool, you will not notice this order, but if you iterate over the pool or save it then the internal order will be used. This action allows you to order the numbers in the pool. The **Order** parameter specifies whether to sort in ascending (1) or descending (0) order. The **What to Sort** parameter specified whether you want to sort the numbers in the pool (0), the numbers utilized from the pool (1), or both (2).

Expressions

Get Generated Number (Pool ID)

This expression returns the last generated number from the pool.

Get and Use New Generated Number (Pool ID)

This is the equivalent to the fast number generation in the original Random Pool. It will generate a new number and return it right away. This will also flag the number as utilized.

Get Iterated Number (Pool ID)

When an iteration action is performed, this expression allows you to get the current iterated number.

Get Minimum of Pool (Pool ID)

This expression will return the lower limit of the specified pool.

Get Maximum of Pool (Pool ID)

This expression will return the upper limit of the specified pool.

Get Seed of Pool (Pool ID)

This expression returns the seed used for the specified pool.

Get Number of Numbers Remaining in Pool (Pool ID)

As the name implies clearly, this expression will return the number of numbers remaining in the pool.

Get Number of Numbers Utilized from Pool (Pool ID)

As the name implies clearly, this expression will return the number of numbers utilized in the pool.

Get as String Pool Numbers (Pool ID, Delimiter)

This expression will return the numbers in the pool as a string delimited with the specified delimiter.

Get as String Pool Utilized Numbers (Pool ID, Delimiter)

This expression will return the numbers utilized from the pool as a string delimited with the specified delimiter.