

## Multi-Threading

Multi-Threading plugin is a plugin that is close to the performance plugin. It can create new threads which are just like functions but looped that are performance driven. Meaning it uses the same code that my Game Performance plugin uses. You can run your threads at a rate called Loops Per Second or LPS for short. I also added a resource query which is basically a list of resources to be loaded but on a new thread. You can only have one resource query thread to load objects. Having multiple resource queries or loading a resource at the same time the query is running will cause the game to crash.

However there are a few rules to running things threads: (To avoid crashes)

- Do not load resources while a query is loading resources
- You must not try accessing the same variable from two separate threads at the same time without the use of a mutex. (MatrixUtils have mutexes)
- You may not load a resource within a new thread at the same time a main thread or another thread is loading resources.

Dark Basic Pro is natively not thread safe. This means that there is no guarantee that Dark Basic Pro commands will work successfully in a thread.

What Multi-Threading is useful for:

- Math calculations
- Loading Resources using my resource query while the main loop is running
- Physics
- For loops that are heavy in work or very long for loops that will hinder with the game play
- Input: I created a plugin to handle multiple keys at once.
- And many more!

## Commands

### MT Start

This command will setup the multi-threading so you can use the commands

Index = **MT Create Task**(FunctionName\$, LPSCap)

This command will create a new multi-threaded task. You provide a Function Name that will be called. The LPS cap is the rate in Loops Per Second.

Type\$ = **MT Get Task Type**()

- Time: Will run the task for a certain amount of time
- Loop: Will loop indefinitely
- Count: Will run the task for a certain amount of iterations

### MT End Task

**MT Set Task To Time** Index, Duration#

This command will run a task for a certain amount of time. (Duration#) The Index is the Index of the

already created Task.

#### **MT Set Task To Count** Index, Count

This command will run a task for a certain amount of iterations. (Count) The Index is the Index of the already created Task.

#### **MT Set Task To Loop** Index

This command will run a task indefinitely. The Index is the Index of the already created Task.

-

#### **MT Pause Task** Index

This command will pause the task.

#### **MT Resume Task** Index

This command will resume the pause task.

#### **MT Stop Task** Index

This command will stop the task.

index = **MT Task Count**()

This command will return the amount of tasks that exists.

index = **MT Task Ended**()

This command will return a 1 if the task has ended and a 0 if it hasn't.

#### **MT Remove Task** Index

This command will remove a task at Index.

#### **MT Query Object** ObjectIndex, Path\$, PositionVector, RotateVector, Exclude Flag

This command will add an object to the query. You provide a Path\$ and an Object Index. The Position and Rotate Vectors are DBP vectors with the position and rotations set. The Exclude flag set to 1 will exclude the object as soon as it is loaded and a 0 won't exclude it.

#### **MT Query Image** Index, Path\$

This command will add a image to the query. You provide a Path\$ and an Object Index.

#### **MT Query Sound** Index, Path\$

This command will add a sound to the query. You provide a Path\$ and an Object Index.

**MT Query Music** Index, Path\$

This command will add music to the query. You provide a Path\$ and an Object Index.

ThreadIndex = **MT Load Resources**()

This command will create a new thread where it will go through the queried resources and load them one at a time.