Details

Igor starts up with sleepTicks = 1. Use 0 to give Igor maximum time, use a larger number to give other applications more time.

Background tasks are used mainly by data acquisition programs.

See Also

Background Tasks on page IV-319 and the **SetBackground** operation.

SetRandomSeed

`SetRandomSeed` ***`seed`***

The SetRandomSeed operation seeds the random number generator used for the noise functions listed under **Noise Functions** on page III-390.

Use SetRandomSeed if you need "random" numbers that are reproducible. If you don't use SetRandomSeed, the random number generator is initialized using the system clock when Igor starts. This almost guarantees that you will never get the same sequence twice unless you use SetRandomSeed.

Flags

/BETR[=better]	If better is absent or non-zero, a better method is used for seeding the Mersenne Twister random number generator. /BETR is ignored for all other random generators.

Parameters

seed should be a number in the interval (0, 1]. For any given *seed*, enoise or gnoise or any of the other random-number generator functions generates a particular sequence of pseudorandom numbers. Calling SetRandomSeed with the same seed restarts and repeats the sequence.

Details

Igor's noise functions are listed under **Noise Functions** on page III-390. The enoise and gnoise functions allow you to choose a random number generator. The other functions always use the Mersenne Twister generator.

How the seed is used internally depends on the generator. For the Linear Congruential Generator the seed is scaled to a 32-bit signed integer. For the Mersenne Twister the seed is scaled to a 32-bit unsigned integer. Both only use the lower 16-bits of the so scaled value for historic reasons.

The Xoshiro256** generator uses all available bits and scales it to an unsigned 64-bit integer.

All generators use the scaled seed value when initializing their internal state tables.

See Also

The **enoise** and **gnoise** functions. **Noise Functions** on page III-390.

SetScale

`SetScale` `[`**`/I/P`**`]` ***`dim`***`,` ***`num1`***`,` ***`num2`*** `[,` ***`unitsStr`***`],` ***`waveName`*** `[,` ***`waveName`***`]`…

`SetScale` ***`d`***`,` ***`num1`***`,` ***`num2`*** `[,` ***`unitsStr`***`],` ***`waveName`*** `[,` ***`waveName`***`]`…

The SetScale operation sets the dimension scaling or the data full scale for the named waves.

Parameters

The first parameter *dim* must be one of the following:

Character	Signifies
d	Data full scale.
t	Scaling of the chunks dimension (t scaling).
x	Scaling of the rows dimension (x scaling).
y	Scaling of the columns dimension (y scaling).
z	Scaling of the layers dimension (z scaling).