Ice Thread Pool Properties

A communicator creates two thread pools: the client thread pool dispatches AMI callbacks and incoming requests on bidirectional connections, and the server thread pool dispatches requests to object adapters.

This page describes configuration properties for the client and server thread pools. These thread pools are named `Client` and `Server`, respectively. In the property descriptions below, replace *name* with `Client` or `Server`.

On this page:

- Ice.ThreadPool.name.Serialize
- Ice.ThreadPool.name.Size
- Ice.ThreadPool.name.SizeMax
- Ice.ThreadPool.name.SizeWarn
- Ice.ThreadPool.name.StackSize
- Ice.ThreadPool.name.ThreadIdleTime
- Ice.ThreadPool.name.ThreadPriority
- Ice.ThreadPriority

Ice.ThreadPool.*name*.Serialize

Synopsis

`Ice.ThreadPool.`*name*`.Serialize=`*num*

Description

If *num* is a value greater than zero, the `Client` or `Server` thread pool serializes all messages from each connection. It is not necessary to enable this feature in a thread pool whose maximum size is one thread. In a multi-threaded pool, enabling serialization allows requests from different connections to be dispatched concurrently while preserving the order of messages on each connection. Note that serialization has a signficant impact on latency and throughput. If not defined, the default value is zero.

Ice.ThreadPool.*name*.Size

Synopsis

`Ice.ThreadPool.`*name*`.Size=`*num*

Description

Thread pools in Ice can grow and shrink dynamically, based on an average load factor. A thread pool always has at least one thread and may grow as load increases up to the maximum size specified by `Ice.ThreadPool.`*name*`.SizeMax`. If `SizeMax` is not specified, Ice uses the value of *num* as the pool's maximum size. The `Client` or `Server` thread pool is initialized with *num* active threads, but the pool may shrink to only one thread during idle periods as determined by `Ice.ThreadPool.`*name*`.ThreadIdleTime`.

If not specified, the default value is one for both properties.

An object adapter can also be configured with its own thread pool.

Note that multiple threads for the client thread pool are only required for nested AMI invocations, or to allow multiple AMI callbacks to be dispatched concurrently.

To monitor the thread pool activities of the Ice run time, enable the `Ice.Trace.ThreadPool` property.

Ice.ThreadPool.*name*.SizeMax

Synopsis

`Ice.ThreadPool.`*name*`.SizeMax=`*num*

Description

num is the maximum number of threads for the `Client` or `Server` thread pool. Refer to the `Ice.ThreadPool.`*name*`.Size` property for more information on configuring the size of a thread pool.

The default value for `SizeMax` is the value of `Size`, meaning the thread pool can never grow larger than its initial size.

To monitor the thread pool activities of the Ice run time, enable the `Ice.Trace.ThreadPool` property.

Ice.ThreadPool.*name*.SizeWarn

Synopsis

`Ice.ThreadPool.`*name*`.SizeWarn=`*num*

Description

Whenever *num* threads are active in the `Client` or `Server` thread pool, a "low on threads" warning is printed. The default value is zero, which disables the warning.

To monitor the thread pool activities of the Ice run time, enable the `Ice.Trace.ThreadPool` property.

Ice.ThreadPool.*name*.StackSize

Synopsis

`Ice.ThreadPool.`*name*`.StackSize=`*num*

Description

num is the stack size (in bytes) of threads in the `Client` or `Server` thread pool. The default value is zero, meaning the operating system's default is used.

Ice.ThreadPool.*name*.ThreadIdleTime

Synopsis

`Ice.ThreadPool.`*name*`.ThreadIdleTime=`*num*

Description

Ice can automatically reap idle threads in the `Client` or `Server` thread pool to conserve resources. This property specifies the number of seconds a thread must be idle before it is reaped. If not specified, the default value is 60 seconds.

To disable the reaping of idle threads, set `ThreadIdleTime` to zero. In this situation, the thread pool is initialized with `Ice.ThreadPool.`*name*`.Size` active threads and may grow to contain `Ice.ThreadPool.`*name*`.SizeMax` active threads, but the size of the pool never decreases.

To monitor the thread pool activities of the Ice run time, enable the `Ice.Trace.ThreadPool` property.

Ice.ThreadPool.*name*.ThreadPriority

Synopsis

`Ice.ThreadPool.`*name*`.ThreadPriority=`*num*

Description

num specifies a thread priority for the threads in the `Client` or `Server` thread pool. Leaving this property unset causes the run time to create threads with the default priority specified by `Ice.ThreadPriority`.

This property is unset by default.

You can also override the default priority for a specific object adapter using `adapter.ThreadPool.ThreadPriority`.

Ice.ThreadPriority

Synopsis

`Ice.ThreadPriority=num`

Description

`num` specifies a thread priority. Threads created by the Ice run time are created with the specified priority by default. Leaving this property unset causes the run time to create threads with the system default priority. This property is unset by default.

You can separately override the default priorities for the client and server thread pools using `Ice.ThreadPool.name.ThreadPriority` as well as for a specific object adapter using `adapter.ThreadPool.ThreadPriority`.