

Miscellaneous Ice.* Properties

On this page:

- [Ice.BackgroundLocatorCacheUpdates](#)
- [Ice.BatchAutoFlushSize](#)
- [Ice.CacheMessageBuffers](#)
- [Ice.ChangeUser](#)
- [Ice.ClassGraphDepthMax](#)
- [Ice.CollectObjects](#)
- [Ice.Compression.Level](#)
- [Ice.ConsoleListener](#)
- [Ice.EventLog.Source](#)
- [Ice.HTTPProxyHost](#)
- [Ice.HTTPProxyPort](#)
- [Ice.ImplicitContext](#)
- [Ice.LogFile](#)
- [Ice.LogFile.SizeMax](#)
- [Ice.LogStdErr.Convert](#)
- [Ice.MessageSizeMax](#)
- [Ice.Nohup](#)
- [Ice.NullHandleAbort](#)
- [Ice.Package.module](#)
- [Ice.PreloadAssemblies](#)
- [Ice.PrintAdapterReady](#)
- [Ice.PrintProcessId](#)
- [Ice.PrintStackTraces](#)
- [Ice.ProgramName](#)
- [Ice.RetryIntervals](#)
- [Ice.ServerIdleTime](#)
- [Ice.SOCKSProxyHost](#)
- [Ice.SOCKSProxyPort](#)
- [Ice.StdErr](#)
- [Ice.StdOut](#)
- [Ice.SyslogFacility](#)
- [Ice.ThreadInterruptSafe](#)
- [Ice.ToStringMode](#)
- [Ice.UseApplicationClassLoader](#)
- [Ice.UseSyslog](#)

Ice.BackgroundLocatorCacheUpdates

Synopsis

`Ice.BackgroundLocatorCacheUpdates=num`

Description

If *num* is set to 0 (the default), an invocation on an indirect proxy whose endpoints are older than the configured [locator cache](#) timeout triggers a locator cache update; the run time delays the invocation until the new endpoints are returned by the locator.

If *num* is set to a value larger than 0, an invocation on an indirect proxy with expired endpoints still triggers a locator cache update, but the update is performed in the background, and the run time uses the expired endpoints for the invocation. This avoids delaying the first invocation that follows expiry of a cache entry.

[Back to Top](#) ^

Ice.BatchAutoFlushSize

Synopsis

`Ice.BatchAutoFlushSize=num`

Description

This property controls how the Ice run time deals with flushing of [batch messages](#). If *num* is set to a value greater than 0, the run time automatically forces a flush of the current batch when a new message is added to a batch and that message would cause the batch to exceed *num* kilobytes. If *num* is set to 0 or a negative number, batches must be flushed explicitly by the application. If not defined, the default value is 1024.



When flushed, batch requests are sent as a single Ice message. The Ice run time in the receiver limits incoming messages to the maximum size specified by `Ice.MessageSizeMax`, therefore the sender must periodically flush batch requests (whether manually or automatically) to ensure they do not exceed the receiver's configured limit.

[Back to Top ^](#)

Ice.CacheMessageBuffers

Synopsis

`Ice.CacheMessageBuffers=num` (Java, .NET)

Description

If `num` is a value greater than 0, the Ice run time caches message buffers for future reuse. This can improve performance and reduce the amount of garbage produced by Ice internals that the garbage collector would eventually spend time to reclaim. However, for applications that exchange very large messages, this cache may consume excessive amounts of memory and therefore should be disabled by setting this property to 0.



This property affects the caching of message buffers for synchronous invocation, asynchronous invocation, and synchronous dispatch. The Ice run time never caches message buffers for asynchronous dispatch.

Platform Notes

- Java
Ice allocates non-direct message buffers when this property is set to 1 and direct message buffers when set to 2. Use of direct message buffers minimizes copying and typically results in improved throughput. If not defined, the default value is 2.
- .NET
If not defined, the default value is 1.

[Back to Top ^](#)

Ice.ChangeUser

Synopsis

`Ice.ChangeUser=user` (C++ & Unix only)

Description

If set, Ice changes the user and group id to the respective ids of `user` in `/etc/passwd`. This only works if the Ice application is executed by the super-user.

[Back to Top ^](#)

Ice.ClassGraphDepthMax

Synopsis

`Ice.ClassGraphDepthMax=num`

Description

Specifies the maximum depth for a graph of Slice class instances to unmarshal. If this maximum is reached, the Ice run time throws an `Ice::MarshalException`. Reading and destroying a Slice class graph are recursive operations. This property prevents stack overflows from occurring if a sender sends a very large graph and not enough space on the stack is available. To read larger graphs, you can increase the value of this property but you should also make sure the stack size is still large enough or increase it using the [thread pool StackSize](#) property. If not specified, the default value is 100.

If this is not a concern, you can set `Ice.ClassGraphDepthMax` to 0; setting this property to 0 (or to a negative number) disables the depth limit altogether.

[Back to Top ^](#)

Ice.CollectObjects

Synopsis

`Ice.CollectObjects=num` (C++98)

Description

Ice for C++ includes a [garbage collection facility](#) for reclaiming cyclic graphs of Slice class instances that are unmarshaled by the Ice run time. Setting this property to 1 causes the Ice run time to assume that all cyclic object graphs that a program receives are eligible for collection by default. If not specified, the default value is 0.

[Back to Top](#) ^

Ice.Compression.Level

Synopsis

`Ice.Compression.Level=num`

Description

Specifies the bzip2 compression level to use when [compressing protocol messages](#). Legal values for *num* are 1 to 9, where 1 represents the fastest compression and 9 represents the best compression. Note that higher levels cause the bzip2 algorithm to devote more resources to the compression effort, and may not result in a significant improvement over lower levels. If not specified, the default value is 1.

[Back to Top](#) ^

Ice.ConsoleListener

Synopsis

`Ice.ConsoleListener=num` (.NET)

Description

If *num* is non-0, the Ice run time installs a `ConsoleTraceListener` that writes its messages to `stderr`. If *num* is 0, logging is disabled. Note that the setting of [#Ice.LogFile](#) overrides this property: if `Ice.LogFile` is set, messages are written to the log file regardless of the setting of `Ice.ConsoleListener`.

[Back to Top](#) ^

Ice.EventLog.Source

Synopsis

`Ice.EventLog.Source=name` (C++ & Windows only)

Description

Specifies the name of an event log source to be used by a Windows service that subclasses [Ice::Service](#). The value of *name* represents a subkey of the `Eventlog` registry key. An application (or administrator) typically prepares the registry key when the service is installed. If no matching registry key is found, Windows logs events in the `Application` log. Any backslashes in *name* are silently converted to forward slashes. If not defined, `Ice::Service` uses the service name as specified by the `--service` option.

[Back to Top](#) ^

Ice.HTTPProxyHost

Synopsis

`Ice.HTTPProxyHost=addr`

Description

Specifies the host name or IP address of an HTTP proxy server. If *addr* is not empty, Ice uses the designated HTTP proxy server for all outgoing (client) connections.

[Back to Top ^](#)

Ice.HTTPProxyPort

Synopsis

```
Ice.HTTPProxyPort=num
```

Description

The port number of the HTTP proxy server. If not specified, the default value is 1080.

[Back to Top ^](#)

Ice.ImplicitContext

Synopsis

```
Ice.ImplicitContext=type
```

Description

Specifies whether a communicator has an [implicit request context](#) and, if so, at what scope the context applies. Legal values for this property are `None` (equivalent to the empty string), `PerThread`, and `Shared`. If not specified, the default value is `None`.



The `PerThread` type is currently not available for JavaScript or with C++ on Universal Windows (UWP).

[Back to Top ^](#)

Ice.LogFile

Synopsis

```
Ice.LogFile=file
```

Description

Replaces the communicator's [default logger](#) with a simple file-based logger implementation. This property does not affect the [per-process logger](#). The logger creates the specified file if necessary, otherwise it appends to the file. If the logger is unable to open the file, the application receives an `InitializationException` during [communicator initialization](#). If a logger object is supplied in the `InitializationData` argument during communicator initialization, it takes precedence over this property.

[Back to Top ^](#)

Ice.LogFile.SizeMax

Synopsis

```
Ice.LogFile.SizeMax=num (C++)
```

Description

num is a positive integer that represents the maximum size of log files configured through `Ice.LogFile`, in bytes. When a log file's size reaches *num*, the Ice file-based logger renames this log file to `baselogfilename-datetimestamp.ext` and creates a new log file. The default value for *num* is 0, which means that the log file's size is unlimited. In this case, the Ice file-based logger opens and writes to a single log file.

[Back to Top ^](#)

Ice.LogStdErr.Convert

Synopsis

`Ice.LogStdErr.Convert=num` (C++)

Description

If *num* is set to a value larger than 0, on Windows, the communicator's [default logger](#) converts log messages from the application's narrow string encoding (as defined by the installed narrow [string converter](#), if any) to the Windows console's code page. The default value for this property is 1 when `Ice.StdErr` is not set, and 0 otherwise. This property is read by the first communicator created in a process; it is ignored by other communicators.

[Back to Top ^](#)

Ice.MessageSizeMax

Synopsis

`Ice.MessageSizeMax=num`

Description

This property controls the maximum size (in kilobytes) of an uncompressed protocol message that is accepted by the Ice run time. The size includes the size of the Ice protocol header. The default size is 1024 (1 megabyte).

The only purpose of this property is to prevent a malicious or defective sender from triggering a large memory allocation in a receiver. If this is not a concern, you can set `Ice.MessageSizeMax` to 0; setting this property to 0 (or to a negative number) disables the message size limit altogether.

If the Ice run time in a receiver encounters an incoming message whose size exceeds the receiver's setting for `Ice.MessageSizeMax`, the run time raises a `MemoryLimitException` and closes the connection. For example, when a client receives an oversized reply message, the result of its invocation is a `MemoryLimitException`. When a server receives an oversized request message, the client receives a `ConnectionLostException` (because the server closed the connection) and the server logs a message if `Ice.Warn.Connections` is set.

[Back to Top ^](#)

Ice.Nohup

Synopsis

`Ice.Nohup=num` (C++, .NET)

Description

If *num* is set to a value larger than 0, the `Application` convenience class (as well as the `Ice::Service` class in C++) ignore `SIGHUP` on Unix and `CTRL_LOGOFF_EVENT` on Windows. As a result, an application that sets `Ice.Nohup` continues to run if the user that started the application logs off. The default value for `Application` is 0, and the default value for `Ice::Service` is 1.

[Back to Top ^](#)

Ice.NullHandleAbort

Synopsis

`Ice.NullHandleAbort=num` (C++, Objective-C, PHP, Python, Ruby)

Description

If *num* is set to a value larger than 0, invoking an operation using a null [Ice C++98 smart pointer](#) causes the program to abort immediately instead of raising `IceUtil::NullHandleException`.



With the C++11 mapping, these smart pointers are used only in Ice-internal code. This property has no effect on `std::shared_ptr`.

[Back to Top ^](#)

Ice.Package.module

Synopsis

`Ice.Package.module=package` (Java, MATLAB)

Description

Ice for Java and Ice for MATLAB allow you to customize the packaging of generated code. If you use this feature, the Ice run time requires additional configuration in order to successfully unmarshal exceptions and concrete class types. This property associates a top-level Slice *module* with a Java or MATLAB *package*. If all top-level modules are generated into the same user-defined package, it is easier to use [Ice.Default.Package](#) instead.

[Back to Top ^](#)

Ice.PreloadAssemblies

Synopsis

`Ice.PrintProcessId=num` (.NET)

Description

If *num* is set to a value larger than 0, the Ice run-time will try to load all the assemblies referenced by the process during communicator initialization, otherwise the referenced assemblies will be initialized when the Ice run-time needs to lookup a C# class. The default value is 0.

[Back to Top ^](#)

Ice.PrintAdapterReady

Synopsis

`Ice.PrintAdapterReady=num`

Description

If *num* is set to a value larger than 0, an object adapter prints "*adapter_name* ready" on standard output after activation is complete. This is useful for scripts that need to wait until an object adapter is ready to be used.

[Back to Top ^](#)

Ice.PrintProcessId

Synopsis

`Ice.PrintProcessId=num`

Description

If *num* is set to a value larger than 0, the process ID is printed on standard output upon startup.

[Back to Top ^](#)

Ice.PrintStackTraces

Synopsis

`Ice.PrintStackTraces=num` (C++, JavaScript, Objective-C, PHP, Python, Ruby)

Description

(C++) If *num* is set to a value larger than 0, inserting an exception that derives from `Ice::Exception` into a [logger helper class](#) (such as `Ice::Warning`) also displays the exception's stack trace. Likewise, the `ice_stackTrace` function on the base exception class, `Ice::Exception`, will return the stack trace or an empty string depending on the value *num*. If not set, the default value depends on how the Ice run time is compiled: 0 for an optimized build and 1 for a debug build.



When `PrintStackTraces` is non-0, the constructor of `Ice::Exception` captures the current call stack, which adds a small overhead. This overhead is incurred even when the application never converts the captured stack into a stack trace.

On Windows, you need the Ice PDB files to obtain usable stack traces. If you build Ice from sources, the Ice build system always creates PDB files next to your DLLs and executables, and Windows will locate and use these PDB files. If you use Ice NuGet packages, follow the instructions in [Using the Windows Binary Distributions](#).



This property also affects scripting languages that use the Ice for C++ run time (Objective-C, PHP, Python and Ruby). Enabling this property in these languages will only display C++ stack traces.

(JavaScript) If `num` is set to a value larger than 0, `Ice.Exception.toString` includes the exception's stack trace. The default value for `num` is 0.

[Back to Top ^](#)

Ice.ProgramName

Synopsis

```
Ice.ProgramName=name
```

Description

`name` is the program name, which is [set automatically](#) from `argv[0]` (C++) and from `AppDomain.CurrentDomain.FriendlyName` (.NET) during initialization. For Java, `Ice.ProgramName` is initialized to the empty string. The default name can be overridden by setting this property.

[Back to Top ^](#)

Ice.RetryIntervals

Synopsis

```
Ice.RetryIntervals=num [num ...]
```

Description

This property defines the number of times an operation is [automatically retried](#) and the delay between each retry. For example, if the property is set to `0 10 0 500`, the operation is retried 3 times: immediately after the first failure, again after waiting 100ms after the second failure, and again after waiting 500ms after the third failure. The default value (0) means Ice retries once immediately. If set to -1, no retry occurs.

[Back to Top ^](#)

Ice.ServerIdleTime

Synopsis

```
Ice.ServerIdleTime=num
```

Description

If `num` is set to a value larger than 0, Ice automatically calls `Communicator::shutdown` once the communicator has been idle for `num` seconds. This shuts down the communicator's server side and causes all threads waiting in `Communicator::waitForShutdown` to return. After that, a server will typically do some clean-up work before exiting. The default value is 0, meaning that the server will not shut down automatically. This property is often used for servers that are automatically [activated by IceGrid](#).



On Windows, the server idle time takes effect only once the server thread pool idle threads have been reaped (the thread idle time can be configured with the [ThreadIdleTime](#) thread pool property).

[Back to Top ^](#)

Ice.SOCKSProxyHost

Synopsis

`Ice.SOCKSProxyHost=addr`

Description

Specifies the host name or IP address of a SOCKS proxy server. If *addr* is not empty, Ice uses the designated SOCKS proxy server for all outgoing (client) connections.



Ice currently only supports the SOCKS4 protocol, which means only IPv4 connections are allowed.

[Back to Top ^](#)

Ice.SOCKSProxyPort

Synopsis

`Ice.SOCKSProxyPort=num`

Description

The port number of the SOCKS proxy server. If not specified, the default value is 1080.

[Back to Top ^](#)

Ice.StdErr

Synopsis

`Ice.StdErr=filename`

Description

If *filename* is not empty, the standard error stream of this process is redirected to this file, in append mode. This property is checked only for the first communicator that is created in a process.

[Back to Top ^](#)

Ice.StdOut

Synopsis

`Ice.StdOut=filename`

Description

If *filename* is not empty, the standard output stream of this process is redirected to this file, in append mode. This property is checked only for the first communicator created in a process.

[Back to Top ^](#)

Ice.SyslogFacility

Synopsis

`Ice.SyslogFacility=string` (Unix only)

Description

This property sets the syslog facility to *string*. This property has no effect if `Ice.UseSyslog` is not set.

string can be any of syslog facilities: LOG_AUTH, LOG_AUTHPRIV, LOG_CRON, LOG_DAEMON, LOG_FTP, LOG_KERN, LOG_LOCAL0, LOG_LOCAL1, LOG_LOCAL2, LOG_LOCAL3, LOG_LOCAL4, LOG_LOCAL5, LOG_LOCAL6, LOG_LOCAL7, LOG_LPR, LOG_MAIL, LOG_NEWS, LOG_SYSLOG, LOG_USER, LOG_UUCP.

The default value is `LOG_USER`.

[Back to Top ^](#)

Ice.ThreadInterruptSafe

Synopsis

`Ice.ThreadInterruptSafe=num` (Java)

Description

If *num* is set to a value larger than 0, Ice for Java disables message caching by setting `Ice.CacheMessageBuffers` to 0 and takes other steps necessary to ensure that [Java interrupts](#) work correctly. If not defined, the default value is 0.

[Back to Top ^](#)

Ice.ToStringMode

Synopsis

`Ice.ToStringMode=string`

Description

string must be one of the following: `Unicode`, `ASCII`, `Compat`.

This property maps to an enumerator of `Ice::ToStringMode` and controls how `identityToString` and `proxyToString` on the [Communicators](#) local interface escape non-printable ASCII characters and non-ASCII characters.

The default value is `Unicode`.

[Back to Top ^](#)

Ice.UseApplicationClassLoader

Synopsis

`Ice.UseApplicationClassLoader=num` (Java)

Description

If *num* is set to a value larger than 0, when dispatching an operation to a servant, Ice sets the context class loader to the class loader of the servant class. With the Java-Compat mapping only, it also sets the context class loader when calling AMI callbacks (with the class loader of the AMI callback class).

[Back to Top ^](#)

Ice.UseSyslog

Synopsis

`Ice.UseSyslog=num` (Unix only)

Description

If *num* is set to a value larger than 0, a special [logger](#) is installed that logs to the `syslog` service instead of standard error. The identifier for `syslog` is the value of `Ice.ProgramName`. Use `Ice.SyslogFacility` to select a `syslog` facility.

[Back to Top ^](#)