The Per-Process Logger

Previous

Next

Ice allows you to install a per-process custom logger. This logger is used by all communicators that do not have their own specific logger established at the time a communicator is created.

You can set a per-process logger in C++ by calling `Ice::setProcessLogger`, and you can retrieve the per-process logger by calling `Ice::getProcessLogger`:

C++11

```
std::shared_ptr<Ice::Logger> getProcessLogger();
void setProcessLogger(const std::shared_ptr<Logger>&);
```

C++98

```
LoggerPtr getProcessLogger();
void setProcessLogger(const LoggerPtr&);
```

If you call `getProcessLogger` without having called `setProcessLogger` first, the Ice run time installs a default per-process logger. Note that if you call `setProcessLogger`, only communicators created after that point will use this per-process logger; communicators created earlier use the logger that was in effect at the time they were created. (This also means that you can call `setProcessLogger` multiple times; communicators created after that point will use whatever logger was established by the last call to `setProcessLogger`.)

`getProcessLogger` and `setProcessLogger` are language-specific APIs that are not defined in Slice. Therefore, these methods appear in the `com.zeroc.Ice.Util` class (for Java), and the `Ice.Util` class (for Java Compat and C#).

For applications that use the `Application` or `Service` convenience classes and do not explicitly configure a logger, these classes set a default per-process logger that uses the `Ice.ProgramName` property as a prefix for log messages. The `Application` class is described in the server-side language mapping chapters; more information on the `Service` class can be found in The `Ice::Service` Class.

Back to Top ^

See Also

- Custom Loggers
- Communicator Initialization
- Application Helper Class
- Service Helper Class

Previous

Next