

Portable Signal Handling in C++

The `IceUtil::CtrlHandler` class provides a portable mechanism to handle Ctrl+C and similar signals sent to a C++ process. On Windows, `IceUtil::CtrlHandler` is a wrapper for `SetConsoleCtrlHandler`; on POSIX platforms, it handles `SIGHUP`, `SIGTERM` and `SIGINT` with a dedicated thread that waits for these signals using `sigwait`. Signals are handled by a callback function implemented and registered by the user. The callback is a simple function that takes an `int` (the signal number) and returns `void`; it must not throw any exception:

C++

```
namespace IceUtil {

    typedef void (*CtrlHandlerCallback)(int);

    class CtrlHandler {
    public:
        CtrlHandler(CtrlHandlerCallback = 0);
        ~CtrlHandler();

        void setCallback(CtrlHandlerCallback);
        CtrlHandlerCallback getCallback() const;
    };
}
```

The member functions of `CtrlHandler` behave as follows:

- `CtrlHandler`
Constructs an instance with a callback function. Only one instance of `CtrlHandler` can exist in a process at a given moment in time. On POSIX platforms, the constructor masks `SIGHUP`, `SIGTERM` and `SIGINT`, then starts a thread that waits for these signals using `sigwait`. For signal masking to work properly, it is imperative that the `CtrlHandler` instance be created before starting any thread, and in particular before initializing an Ice communicator.
- `~CtrlHandler`
Destroys the instance, after which the default signal processing behavior is restored on Windows (`TerminateProcess`). On POSIX platforms, the "sigwait" thread is terminated and joined, but the signal mask remains unchanged, so subsequent signals are ignored.
- `setCallback`
Sets a new callback function.
- `getCallback`
Gets the current callback function.

It is legal specify a value of zero (0) for the callback function, in which case signals are caught and ignored until a non-zero callback function is set.

A typical use for `CtrlHandler` is to shutdown a communicator in an Ice server. For example, the `Ice::Application` class uses a `CtrlHandler` in its implementation.

See Also

- [The Server-Side main Function in C++](#)