these extensions in a syntactically consistent manner. ~~If an unknown attribute is encountered during compilation, it is ignored, emitting~~ a likely[4] nonfatal diagnostic.

Table 1 provides several examples of popular compiler-specific attributes that have been standardized or have migrated to the standard syntax. For additional compiler-specific attributes, see *Further Reading* on page 20.

**Table 1: Some standardized compiler-specific attributes**

| Compiler | Compiler-Specific | Standard-Conforming |
|---|---|---|
| GCC | `__attribute__((pure))` | `[[gnu::pure]]` |
| Clang | `__attribute__((no_sanitize))` | `[[clang::no_sanitize]]` |
| MSVC | `declspec(deprecated)` | `[[deprecated]]` |

Portability is the biggest advantage of preferring standard syntax when it is available for compiler- and external-tool-specific attributes. Because most compilers will simply ignore unknown attributes that use standard attribute syntax (and, as of C++17, they are required to do so), conditional compilation is no longer required.

## Use Cases

### Prompting useful compiler diagnostics

Decorating entities with certain attributes can give compilers enough additional context to provide more detailed diagnostics; e.g., the GCC-specific `[[gnu::warn_unused_result]]` attribute[5] can be used to inform the compiler and developers that a function's return value should not be ignored[6]:

```
struct UDPListener
{
    [[gnu::warn_unused_result]] int start();
        // Start the UDP listener's background thread, which can fail for a
        // variety of reasons. Return 0 on success and a nonzero value
        // otherwise.

    void bind(int port);
        // The behavior is undefined unless start was called successfully.
};
```

[4] Prior to C++17, a conforming implementation was permitted to treat an unknown attribute as ill formed and terminate translation; to the authors' knowledge, however, none of them did.

[5] For compatibility with GCC, Clang supports `[[gnu::warn_unused_result]]` as well.

[6] The C++17 Standard `[[nodiscard]]` attribute serves the same purpose and is portable.