Notes

The wording changes proposed in this paper address national body comment US 40 (Core issue 814). The changes are against N2914.

Attribute [[nothrow]] has much in common with attribute [[noreturn]], but the proposed wording is somewhat different. If this wording for [[nothrow]] is found suitable, perhaps the wording for [[noreturn]] should be made to match.

Wording Changes

Add the following subsection after 7.6.3 [dcl.attr.noreturn]:

7.6.4 Nothrow attribute [dcl.attr.nothrow]

1 The attribute-token noexcept specifies that a function will not terminate by throwing an exception. It shall appear at most once in each attribute-list and shall not have an associated attribute-argument-clause.

2 The attribute shall appertain to a function, member function, function template or member function template. If the attribute appertains to a template, it also applies to every specialization of that template.

3 If an entity is declared with the noexcept attribute, the first declaration of that entity in that translation unit shall specify the noexcept attribute. If an entity is declared noexcept in one translation unit, every translation unit in which the entity is declared shall declare it noexcept, no diagnostic required.

4 If a member function overrides a virtual base-class member function declared noexcept, the overriding function shall also be declared noexcept.

5 If a call to an entity declared noexcept terminates with an exception, the behavior of the program is undefined.

6 [Example:

    [[nothrow]] void f1(); // OK
    void f1 [[nothrow]] () ; // OK
    [[nothrow]] void f1 [[nothrow]] () ; // OK
void e1 [[nothrow, nothrow]]();     // error: cannot repeat attribute-
// token

void [[nothrow]] e2();           // error: the attribute appertains
// to void, not to the function

void e3();
[[nothrow]] void e3();           // error: first declaration doesn't
// have the attribute

[[nothrow]] void u1() {
    throw "bait";
}

int main() {
    u1();                          // undefined behavior because
    // an exception is thrown
}

struct B1 { virtual void f(); };  
struct B2 { [[nothrow]] virtual void f(); };
struct D: B1, B2 {
    void f();                     // error: [[nothrow]] required
};

—end example

7 [ Note: The noexcept attribute does not affect the type of a declaration. It is meant
primarily as a hint for code generators. In particular, when the attribute is combined
with an exception specification a code generator can omit the implicit try/catch
construct usually needed when implementing exception specifications. —end note ]