Unspecialized `std::tuple_size` should be defined

Document number: N4386
Date: 2015-02-24
Project: Programming Language C++, Library Working Group
Reply-to: Nevin "☺" Liber mailto:nliber@drw.com

Introduction

This paper is wording for issue LWG 2446 / LEWG 42.

Discussion

In N4296 [tuple.general] paragraph 2, the unspecialized `std::tuple_size` is undefined. It would be a lot more useful with SFINAE if it were defined as an empty `struct`; that way, it can be used with `enable_if` for determining whether or not it is valid to use `tuple_size`, `tuple_element` and `get` on the corresponding data structure.

Proposed resolution (relative to N4296)

Change 20.4.1 [tuple.general] p2, header <tuple> synopsis, as indicated

```
[…]
// 20.4.2.5, tuple helper classes:
template <class T> class tuple_size; ~~// undefined~~
[…]
```

Change 20.4.2.5 [tuple.helper] as indicated

```
[…]
template <class T> struct tuple_size { };
```

Remarks: All specializations of `tuple_size<T>` shall meet the `UnaryTypeTrait` requirements (20.10.1) with a `BaseCharacteristic` of `integral_constant<size_t, N>` for some `N`.

In addition to being available via inclusion of the `<tuple>` header, the primary template definition is available when either of the headers `<array>` or `<utility>` are included.

```
[…]
```