

`std::false_type`

`std::true_type`

`detail::is_complete
_type< T, decltype(void
(sizeof(T)))>`

```
graph BT; A["detail::is_complete_type< T, decltype(void (sizeof(T)))>"] --> B["std::false_type"]; A --> C["std::true_type"];
```

The diagram illustrates a C++ template specialization. At the bottom, a gray box contains the code for a specialization of `detail::is_complete_type` for a type `T` where `sizeof(T)` is known. Two blue arrows point from this box to two white boxes above it. The left box is labeled `std::false_type` and the right box is labeled `std::true_type`, indicating that the specialization inherits from one of these two types.