Subject: Why there is no Index::Add(T&&)?
Posted by busiek on Sun, 17 Dec 2017 04:42:38 GMT
View Forum Message <> Reply to Message

I need to store large objects as keys in Index, thus I prefer to pass ownership of a key to the container. I had to create my own container:

```
template <class T>
class HeavyIndex : public Index<T>, public MoveableAndDeepCopyOption<HeavyIndex<T>>
{
 typedef Index<T> B;
public:
 T& Add(T&& x, unsigned _hash)
 {
  T& t = B::key.Add(pick(x));
  B::hash.Add(_hash);
  return t;
 }
 T& Add(T&& x) { return Add(pick(x), B::hashfn(x)); }
 int FindAdd(T&& x, unsigned _hash)
 {
  int i = B::Find(x, _hash);
  if(i >= 0) return i;
  i = B::key.GetCount();
  Add(pick(x), _hash);
  return i;
 }
 int FindAdd(T&& x) { return FindAdd(pick(x), B::hashfn(x)); }
};
```

However, probably a single version of Add() method can be created using a solution with std::forward similarly as in Fixes to Array::Create & Vector::Create.