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.

Subject: Re: Why there is no Index::Add(T&&)?
Posted by Novo on Sun, 17 Dec 2017 21:09:38 GMT
View Forum Message <> Reply to Message

busiek wrote on Sat, 16 December 2017 23:42However, 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.
In order to make std::forward work method Add has to be a template method.

```
template<typename A>
A& Add(A&& x, unsigned _hash);
```

Using T&& with a method of a template class won't create a forwarding reference.

Subject: Re: Why there is no Index::Add(T&&)?
Posted by busiek on Sun, 17 Dec 2017 21:19:26 GMT
View Forum Message <> Reply to Message

Novo wrote on Sun, 17 December 2017 22:09
Using T&& with a method of a template class won't create a forwarding reference.

Right. I shouldn't write messages in the middle of the night. :roll:

Subject: Re: Why there is no Index::Add(T&&)?
Posted by mirek on Sun, 28 Jan 2018 18:14:02 GMT
View Forum Message <> Reply to Message

Now in trunk, r-value variants added to all relevant Index, VectorMap, ArrayMap methods.

Subject: Re: Why there is no Index::Add(T&&)?
Posted by busiek on Mon, 07 Jan 2019 13:47:59 GMT
View Forum Message <> Reply to Message

That was great addition. Thanks.

Similar problem is with InVector and relevant containers. Methods like Add, Insert, InsertUpperBound, InsertRange, AppendRange and so on should have T&& variants too. It shouldn't be too much work to add that.

Subject: Re: Why there is no Index::Add(T&&)?
Posted by mirek on Tue, 08 Jan 2019 08:36:18 GMT
View Forum Message <> Reply to Message

Issue filed, will happen.

Mirek