
Subject: Re: get_i
Posted by [mirek](#) on Wed, 17 Jun 2020 11:03:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

So I started looking into eliminating all instances of Expand macro usage and identified that following helpers could be quite useful:

```
template <class I, class V>
void iter_set(I t, V&& v)
{
    *t++ = v;
}
```

```
template <class I, class V, typename... Args>
void iter_set(I t, V&& v, Args&& ...args)
{
    *t++ = v;
    iter_set(t, args...);
}
```

```
template <class C, typename... Args>
C gather(Args&& ...args)
{
    C x;
    x.SetCount(sizeof...(args));
    iter_set(x.begin(), args...);
    return x;
}
```

```
template <class I, class V>
void iter_get(I s, V& v)
{
    v = *s++;
}
```

```
template <class I, class V, typename... Args>
void iter_get(I s, V& v, Args& ...args)
{
    v = *s++;
    iter_get(s, args...);
}
```

```
template <class C, typename... Args>
int scatter(int n, const C& c, Args& ...args)
{
    if(n < sizeof...(args))
        return 0;
}
```

```

iter_get(c.begin(), args...);
return sizeof...(args);
}

template <class C, typename... Args>
int scatter(const C& c, Args& ...args)
{
    return scatter(c.GetCount(), c, args...);
}

```

Usage example:

```

template <typename... Args>
String Format(const char *fmt, const Args& ...args)
{
    return Format(fmt, gather<Vector<Value>>>(args...));
}

```

But I guess this would work even better if containers interface was amended to be more "std" (Vector::Vector(int count), size() synonyme for GetCount), so I guess that needs a bit more work...

Mirek
