



[Home](#) > [Articles](#)

Complex Support with Array Building Blocks

[Submit New Art](#)

Let's have a regular C++ array of complex:

```
std::vector<std::complex<float>> vf;
```

Then, let's have an ArBB dense of complex:

```
dense<std::complex<f32>> af;
```

The naïve way of binding 'af' to 'vf' is:

```
bind(af,&vf[0],vf.size());
```

But this will *not* work. If we look up the template definition of bind, it says the first parameter of bind must be an ArBB dense container of some ArBB scalar type, for example, dense<T>. Then, the second parameter of bind must be a pointer to uncaptured<T>

If T is f32, then uncaptured<T> is float

If T is i64, then uncaptured<T> is int64_t, and so on.

But then how do you bind a series of complex values to an ArBB dense complex? In many cases complex is represented as a pair of two floats.

However if T is std::complex<f32> then uncaptured<T> is NOT std::complex<float> because in this case, both dense<T> and uncaptured<T> are of std::complex type. Another way to understand this is that there does not exist an uncaptured<T> for std::complex<f32>.

How does one do the binding then?

std::complex type is not treated as a 'scalar' type in ArBB. Therefore, the intuitive version of bind() does not work for it. We have to use two C/C++ arrays, one for the "real" part and the other for the "imag" part. Then, we have to pass pointers to both arrays to bind().

Full Solution:

```
typedef std::complex<f32> ArBBComplex;
```

```
std::vector<float> f_real(SIZE);
```

```
std::vector<float> f_imag(SIZE);
```

```
// fill in some test data
std::fill(f_real.begin(), f_real.end(), 1.0);
std::fill(f_imag.begin(), f_imag.end(), -1.0);
```

```
dense<ArBBComplex> a_f;
bind(a_f, SIZE, &f_real[0], &f_imag[0]);
```

So far, this is the best known way to achieve binding. However, our engineering team is working on improving the interface, making it more intuitive and straightforward. Please communicate with us through the forum on any requests or suggestions to improve the binding to complex numbers.

Do you need more help?

Click tags links for related articles

[Search Knowledge Base](#)

[Visit User Forums](#)

[Get other Support options](#)

This article applies to: [Intel® Array Building Blocks Knowledge Base](#)

*Trademarks

©Intel
Corporation