

## Problem

What are the possible template types for DevIOs?

## Solution

You can use only those CORBA types for which an equivalent BACI::R[O|W]-type exists.

Think of this C++-code:

```
class foo: public DevIO< CORBA::Octet >
{
    // bla
};
```

Our characteristic component shall have a Baci-property of type long:

```
SmartPropertyPointer< baci::ROlong > bar;
DevIO< CORBA::Octet >* myDevIO;
```

We instantiate the new DevIO somewhere in the initialisation step of the component and pass this to the constructor of baci::ROlong:

```
myDevIO = new foo;
bar = new baci::ROlong(propName, getComponent(), myDevIO);
```

The compiler will create an error for the call of the constructor. It is expected to pass a DevIO< CORBA::Long >\* and not DevIO< CORBA::Octet >\*. This is due to the fact that baci::ROlong is actually a typedef (\$ACSRROOT/include/baciROlong.h):

```
typedef ROcontImpl<ACS_RO_T(long, CORBA::Long)> ROlong;
```

## Related articles

- [How can more people do development with ACS on the same machine without disturbing each other?](#)
- [Which ports are used by ACS?](#)
- [Problems connecting to ACS servers on a remote machine: bad /etc/hosts](#)
- [Why does the getComponent method of ZLegacy/ACS.ContainerServices return an object of type None?](#)
- [Why are some of my print statements not showing up in the container output section of acscommandcenter?](#)