

Problem

I know events are being sent but they cannot be received by Consumer classes. What's wrong?

Solution

There are several possibilities here:

- the *Consumer* is not subscribed to the correct channel
- the *Consumer* is not subscribed to the correct event type
 - subscribe to event type `null` to get all events on that channel
 - this is always necessary in Java for the just partially supported case of receiving a sequence of IDL structs in one event.
- the *Consumer* has not had the *consumerReady* method invoked yet
- the *Consumer* is suspended. Invoke the *resume* method to fix this

Another possibility is a little known problem which could be coming from the *Supplier* itself. When **each field of an ICD-style ALMA event is not explicitly set**, there is a small chance that the event can be sent from the *Supplier* without any exceptions/warnings occurring. As if it's not bad enough that malformed events can be sent without error, what's even worse is that under certain conditions one type of ORB can receive the events flawlessly while another ORB cannot. At present, we know the following circumstances can lead to this bug:

- an enumeration value within an event which is not explicitly set from a C++ *Supplier*
- any field of an event which is equal to *null* sent by a Java *Supplier*

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?](#)