

Creating Custom MXBean

Custom JMX MBeans can be added to Geode members. They can be federated but not aggregated in the JMX manager. This means individual members MBean data can be retrieved from the JMX manager but the values from all the members cannot be combined in one bean.

The steps to create and register a new Geode JMX MBean are:

1. Define MXBean interface
2. Implement MXBean interface with MBean class
3. Register the MBean
4. Federate the MBean

Define MXBean interface

The MXBean interface provides the JMX bean's attributes and operations.

Define MXBean Interface

```
public interface CustomMXBean {
    public int getCustomInt();
    public String getCustomString();
    public String[] listCustomStrings();
}
```

Implement MXBean interface with MBean class

The MBean class provides the implementation of the MXBean.

Implement MBean Class

```
public class CustomMBean implements CustomMXBean {
    public int getCustomInt() {
        return someInt;
    }
    public String getCustomString() {
        return someString;
    }
    public String[] listCustomStrings() {
        return new String[] {"someString1", "someString2"};
    }
}
```

Register the MBean

Registering the MBean causes the MBean to be available to JMX clients of the member. See below for code sample.

Federate the MBean

Federating the MBean causes the MBean to be registered in the JMX manager's MBeanServer which in turn causes the MBean to be available to JMX clients of the JMX manager. In addition, it causes the current values of the MBean's attributes to be periodically sent to the JMX manager.

Note: The JMX manager requires the CustomMXBean interface to be on its classpath.

Register and Federate MBean

```
// Get the ManagementService
Cache cache = ...;
ManagementService service = ManagementService.getManagementService(cache);

// Create the MBean. The domain specified doesn't matter (as long as there is one). It will be replaced by the
'GemFire' domain.
CustomMXBean bean = new CustomMBean();
ObjectName beanName = ObjectName.getInstance("GemFire:type=Member,name=Custom");

// Register the MBean. The GemFire ObjectName will be something like: GemFire:type=Member,name=Custom,
member=192.168.2.13(68439)<v22>-8504
ObjectName gemfireBeanName = service.registerMBean(bean, beanName);

// Federate the MBean. This causes its values to periodically be sent to the JMX manager.
service.federate(gemfireBeanName, CustomMXBean.class, false);
```

In jconsole, the MBean will be available like:

