Gmbal: Annotations for JMX

Ken Cavanaugh

Introduction

- JMX is a powerful standard for managing systems
- But using standard JDK 5 APIs requires too much code
- Annotations provide a powerful tool for solving this problem
  - Add annotations to existing interfaces and classes
  - Simple register interface at runtime to create MBeans from POJOs
  - Keep annotations simple (avoid annotation bloat)

Why Gmbal (or what’s in a name)?

- Originally simply “JMX Annotations” or JMXA
- That wasn’t a good name for Sun copyright and trademark lawyers
- They wanted a name with GlassFish in it someplace
- So we got “GlassFish MBean Annotation Library”
- Needed a short name too, so “gmbal”.
- Pronouncing this as “gumball” seems a better choice than the alternatives (“gimble”, “gamble”)

Goals

- Use annotations to generate AMX-compliant MBeans from existing classes
- Can be used either standalone or inside GlassFish v3
- Make it trivial to register MBeans
- Construct MBeans from application objects at runtime
  - Easy to retrofit to existing code
- ORB specific uses (not hardwired into framework)
  - Replace old (pre-JMX) monitoring system
  - Make ORB manageable by any management tool (e.g. jconsole)
  - Expose complete ORB state for debugging and optimization use

Why?

- I had a problem to solve:
  - I want to make the operation of the ORB visible:
    - For monitoring and management
    - For debugging
    - For optimization
- MBeans are one of several techniques to use
  - btrace (especially with method annotations) is useful
    - GFv3 probes are similar
  - improved logging is another aspect

Kinds of MBeans

- Kinds of MBeans:
  - Standard MBeans (JavaBeans conventions)
  - MXBeans
  - Dynamic MBeans (Constructed at runtime)
  - Open MBeans (Dynamic with restricted data types)
  - Model MBeans
- Gmbal uses Open MBeans with ModelMBean metadata
  - Want dynamic with standard data types (hence use Open MBeans)
  - Want extensible metadata on JDK 5

Outline

- Describe gmbal approach
  - Examples
  - Details
- Compare with others, especially JSR 255
- Screen captures from ORB with simple JMXTest
- Current status
- Further work

Gmbal API: Annotations

- 8 basic annotations:
  - @ManagedObject and @ManagedData
  - @ManagedAttribute
  - @ManagedOperation and @ParameterNames
  - @NameValue
  - @Description
  - @AMXMetadata
- 5 specialized annotations:
  - @InheritedAttribute and @InheritedAttributes
  - @IncludeSubclass
  - @DescriptorKey and @DescriptorFIelds

Gmbal API: Interface and Factory

- One interface (ManagedObjectManager) that handles registration and related operations.
- One concrete class (ManagedObjectManagerFactory) to obtain ManagedObjectManagers.

Example: ManagedObject

```
@ManagedObject
@Description( "A group of Timers or other TimerGroups, which may be enabled or disabled together" )
public interface TimerGroup extends Controllable {
   @ManagedOperation
   @Description( "Add a new Timer or TimerGroup to this TimerGroup" )
   boolean add( Controllable con ) ;

   @ManagedOperation
   @Description( "Remove a new Timer or TimerGroup from this TimerGroup" )
   boolean remove( Controllable con ) ;
}
```

- @ManagedObject defines an MBean (default type in ObjectName is class name)
- @ManagedOperation defines an MBean operation
- @Description gives a description for the attribute/MBean/operation
- Can also use @ManagedAttribute and a few other annotations

Example: ManagedData

```
@ManagedData(
@Description( "..." )
@InheritedAttribute( id="iterator" )
public interface IOR extends List<TaggedProfile>, Writeable, MakeImmutable {
    ORB getORB() ;

    @ManagedAttribute
    @Description( "..." )
    String getTypeId() ;

    ...
}
```

- @ManagedData indicates that this is mapped to an Open Data Type
- @InheritedAttribute indicates that an attribute named iterator is inherited
  - Also used on ManagedObject methods
- @ManagedAttribute indicates that there is an attribute named typeId
- There is no ORB attribute (we can choose just a subset of the methods)

Example: IncludeSubclass

```
@ManagedObject
@IncludeSubclass( cls = { Timer.class, TimerGroup.class, TimerFactory.class } )
public interface Controllable extends Named { ... }

@ManagedObject
@Description( "A group of Timers or other TimerGroups, which may be enabled or disabled
together" )
public interface TimerGroup extends Controllable {
   @ManagedOperation
   @Description( "Add a new Timer or TimerGroup to this TimerGroup" )
   boolean add( Controllable con ) ;
...
}
```

- Controllable is a base interface that has subclasses Timer, TimerGroup, and TimerFactory
- All Controllables will appear to have an add operation, but it only applies to TimerGroup

ManagedObjectManager API: registration

```
public interface ManagedObjectManager {
    NotificationEmitter getRoot() ;
    NotificationEmitter createRoot() ;
    NotificationEmitter createRoot( Object obj ) ;
    NotificationEmitter createRoot( Object obj, String name ) ;
    NotificationEmitter register( Object obj ) ;
    NotificationEmitter registerAtRoot( Object obj ) ;
    NotificationEmitter registerAtRoot( Object obj, String name ) ;
    NotificationEmitter register( Object parent, Object obj )  ;
    NotificationEmitter register( Object parent, Object obj, String name )  ;
    void unregister( Object obj ) ;
}
```

- Each MOM must have 1 root
- Unregister when needed
- Name required unless object has @NameValue method

ManagedObjectManager API: miscellaneous

```
public interface ManagedObjectManager {
    void suspendJMXRegistration() ;
    void resumeJMXRegistration() ;
    ObjectName getObjectName( Object obj ) ;
    Object getObject( ObjectName oname ) ;
    void stripPrefix( String... str )
    String getDomain() ;
    MBeanServer getMBeanServer() ;
    void setMBeanServer( MBeanServer server ) ;
    ResourceBundle getResourceBundle() ;
    void setResourceBundle( ResourceBundle rb ) ;
    void addAnnotation( AnnotatedElement element, Annotation annotation )
}
```

- Suspend/resume to deal with registration in constructor problem
- Access object name <-> object mapping for registered objects
- set/get MBeanServer and description resource bundle
- stripPrefix and addAnnotations discussed later

ManagedObjectManager API: Debugging Support

```
public interface ManagedObjectManager {
    public enum RegistrationDebugLevel { NONE, NORMAL, FINE }

    void setRegistrationDebugLevel( RegistrationDebugLevel level ) ;
    void setRuntimeDebug( boolean flag ) ;
    void setTypelibDebug( int level ) ;
    String dumpSkeleton( Object obj ) ;
}
```

- Can trace registration of MBeans (construction of skeleton and objectname)
- Can trace runtime operations and get/set on attributes
- Can trace evaluation of types
- Can dump the skeleton of a registered object (shows all metadata for attributes and operations)

ManagedObjectManagerFactory

```
public final class ManagedObjectManagerFactory {
   public static ManagedObjectManager createStandalone( String domain ) { ... }

   public static ManagedObjectManager createFederated( ObjectName rootParentName ) { ... }
}
```

- Only concrete class in the API
- createStandalone(String) takes a domain used in all ObjectNames from this ManagedObjectManager.
  - Typically used in standalone case
- createFederated(rootParentName) takes an AMX-compliant ObjectName which is the parent of the ManagedObjectManager's root
  - Typically used in GFv3 case

Example of setup

```
private void postInit( String[] params, DataCollector dataCollector ) {
   createORBManagedObjectManager() ;
   mom.registerAtRoot( configData ) ;
  ...
}

// from superclass
public void createORBManagedObjectManager() {
   mom = ManagedObjectManagerFactory.createStandalone( "com.sun.corba" ) ;

   if (mbeanFindDebugFlag) {
       mom.setRegistrationDebugFlag( ManagedObjectManager.RegistrationDebugLevel.FINE ) ;
   } ...

   mom.stripPrefix( "com.sun.corba.se", ... ) ;
   mom.createRoot( this, getUniqueOrbId() ) ; // could also just use @NameValue on getUniqueORBId
}
```

- stripPrefix makes default ObjectName types shorter
- Use ORB debug flags to set up gmbal debug
- Need to provide SPI for createFederated call in GFv3

MBean Registration

- This is simple: just call mom.register(someObject) or mom.registerAtRoot(someObject)
  - There are variants for supplying ObjectName name value
- Framework analyzes class of someObject
  - Caches all results of analysis
  - Constructs all required MBean info
  - Constructs TypeConverters that handle conversion between Java types and Open types

Data Mapping Rules

- Rules are very similar to MXBean rules in JDK 6
  - Primitives (along with String, BigDecimal, BigInt, and Date) map to themselves (there is an OpenType for each of these)
  - Anything annotated with @ManagedObject is mapped to its ObjectName
  - @ManagedData is mapped to a CompositeType
  - C<X>, C<? extends X> (C isA Collection, Iterator, Iterable, or Enumeration) is mapped to an array of whatever X maps to
  - M<K,V> (M isA Map or Dictionary) is mapped to TabularData
  - X[] is mapped like C<X>
  - Others map to String (using toString() or <init>(String) if available)

Other Approaches

- JSR 255
- SpoonJMX
- Spring

JSR 255

- Defines many new features (for JDK 7). Some major items include:
  - Namespaces for scalability
  - Event service for enhanced notification support
  - Annotation-driven mbeans (our main concern) and resource injection
    - @MBean/@ManagedAttribute/@ManagedOperation
    - Separate @Description
  - Client contexts for things like client-specific localization
  - A new query language (an alternative to constructing query expressions)

SpoonJMX

- From INRIA in france
- @ManagedResource/@ManagedAttribute/@ManagedOperation
  - @ManagedAttribute includes description and other info
  - @ManagedConstructor for constructing objects
  - @ObjectNameKey for putting fields into ObjectName
  - Implemented as compile-time annotation processor
  - Also uses INRIA's AVAL, a extensible meta-annotation based annotation validator

Spring

- Part of the Spring container (not usable separately as far as I can tell)
- @ManagedResource/@ManagedAttribute/@ManagedOperation
- Also has @ManagedParameter
- Spring has several models for doing this: XML, JavaDoc comments, annotations

Feature Comparison (basic)

Feature	Gmbal	SpoonJMX	Spring	JSR 255
Mark class as MBean	@ManagedObject	@ManagedResource	@ManagedResource	@MBean
Mark class as CompositeData	@ManagedData	N/A	N/A	N/A
Make method as managed attribute	@ManagedAttribute	@ManagedAttribute	@ManagedAttribute	@ManagedAttribute
Make method as managed operation	@ManagedOperation	@ManagedOperation	@ManagedOperation	@ManagedOperation
Mark param as operation parameter	@ParameterName	@ManagedOperation Parameter	@ManagedOperation Parameter	Uses JDK 7 reflection
Mark subclass as variant of superclass	@IncludeSubclass	N/A	N/A	N/A
Inherit an attribute from superclass	@InheritedAttribute(s)	N/A	N/A	N/A

Feature Comparison (advanced)

Feature	Gmbal	SpoonJMX	Spring	JSR 255
Instantiate an MBean	implicit	@ManagedConstructor	implicit?	implicit?
get value for ObjectNameKey	@NameValue	@ObjectNameKey	N/A	N/A
Add a description	@Description	descriptions in other annotations	descriptions in other annotations	@Description
descriptor meta annotations	@AMXMetaData @DescriptorKey @DescriptorFields	N/A	N/A	@DescriptorKey @DescriptorFields
notifications	attribute change only	N/A	N/A	@NotificationInfo(s)
resource injection	might add?	N/A	N/A	@Resource

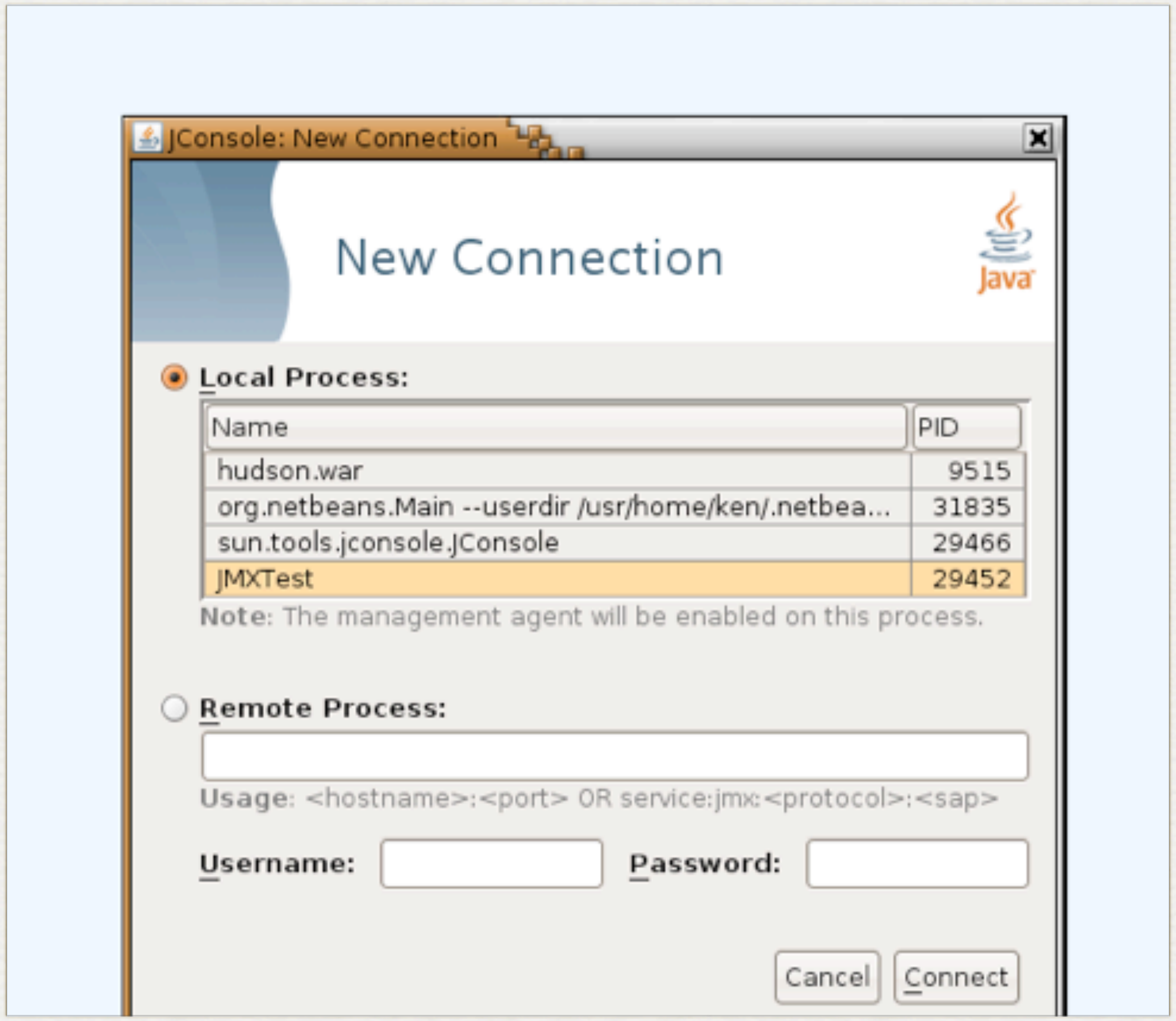
Virtual “Demo”

Simple Test Case

```
@ManagedObject
public class JMXTest {
   public static void main( String[] args ) { (new JMXTest()).run() ; }

   @ManagedOperation
   @Description( "An operation to shutdown this test" )
   public synchronized void shutdown() { notifyAll() ; }

   public synchronized void run() {
      Properties props = new Properties() ;
      props.setProperty( "org.omg.CORBA.ORBClass", "com.sun.corba.ee.impl.orb.ORBImpl" ) ;
      ORB orb = (ORB)ORB.init( (String[])null, props ) ;
      orb.mom().register( this ) ;
      try {
         wait() ;
         System.out.println( "Test is terminating" ) ;
      } catch (Exception exc) {
         exc.printStackTrace() ;
      }
   }
}
```



JConsole: New Connection

New Connection

[✓ Local Process:]

Name	PID
hudson.war	9515
org.netbeans.Main --userdir /usr/home/ken/.netbea...	31835
sun.tools.jconsole.JConsole	29466
JMXTest	29452

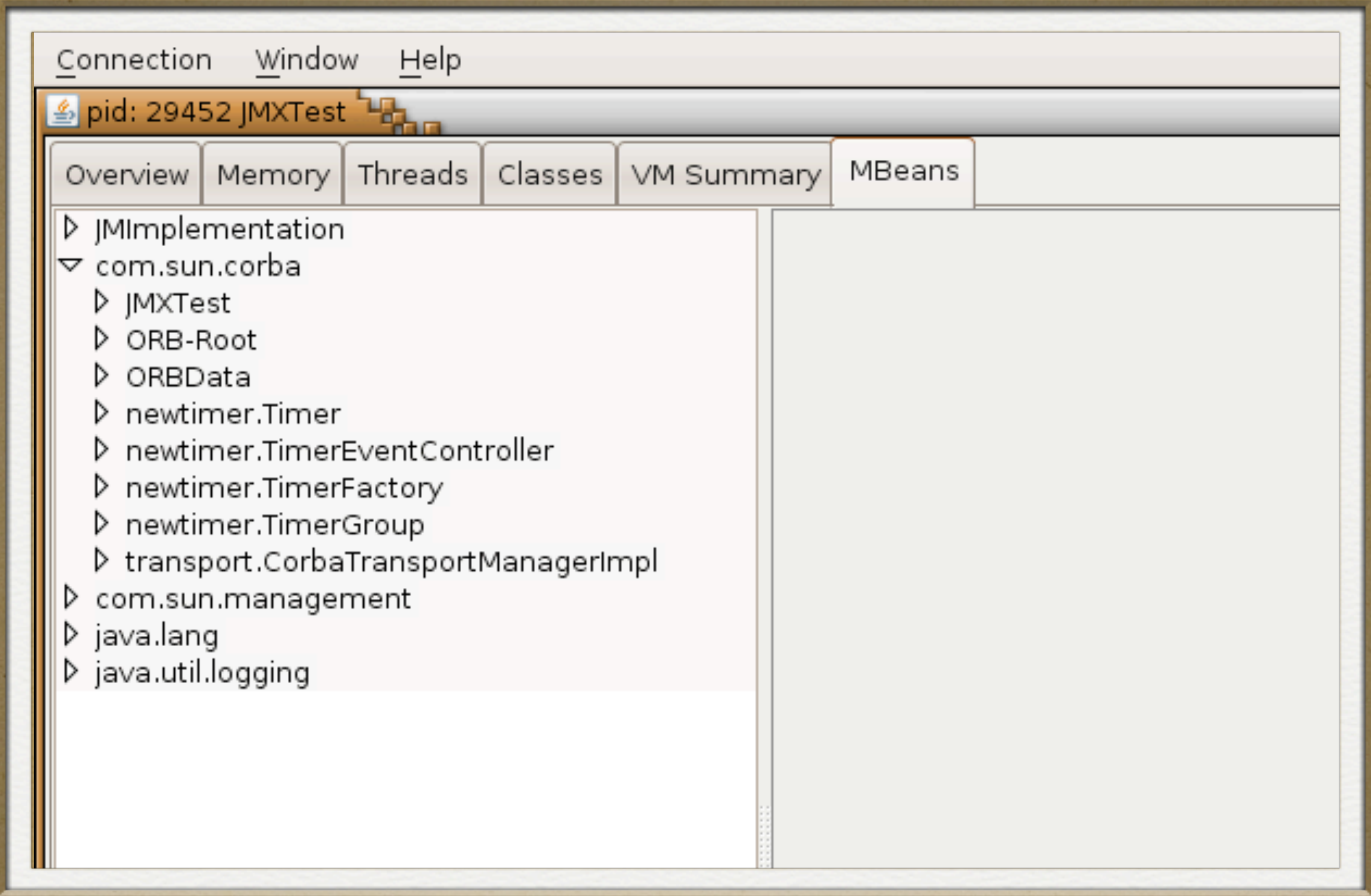
Note: The management agent will be enabled on this process.

Remote Process:

Usage: <hostname>:<port> OR service:jmx:<protocol>:<sap>

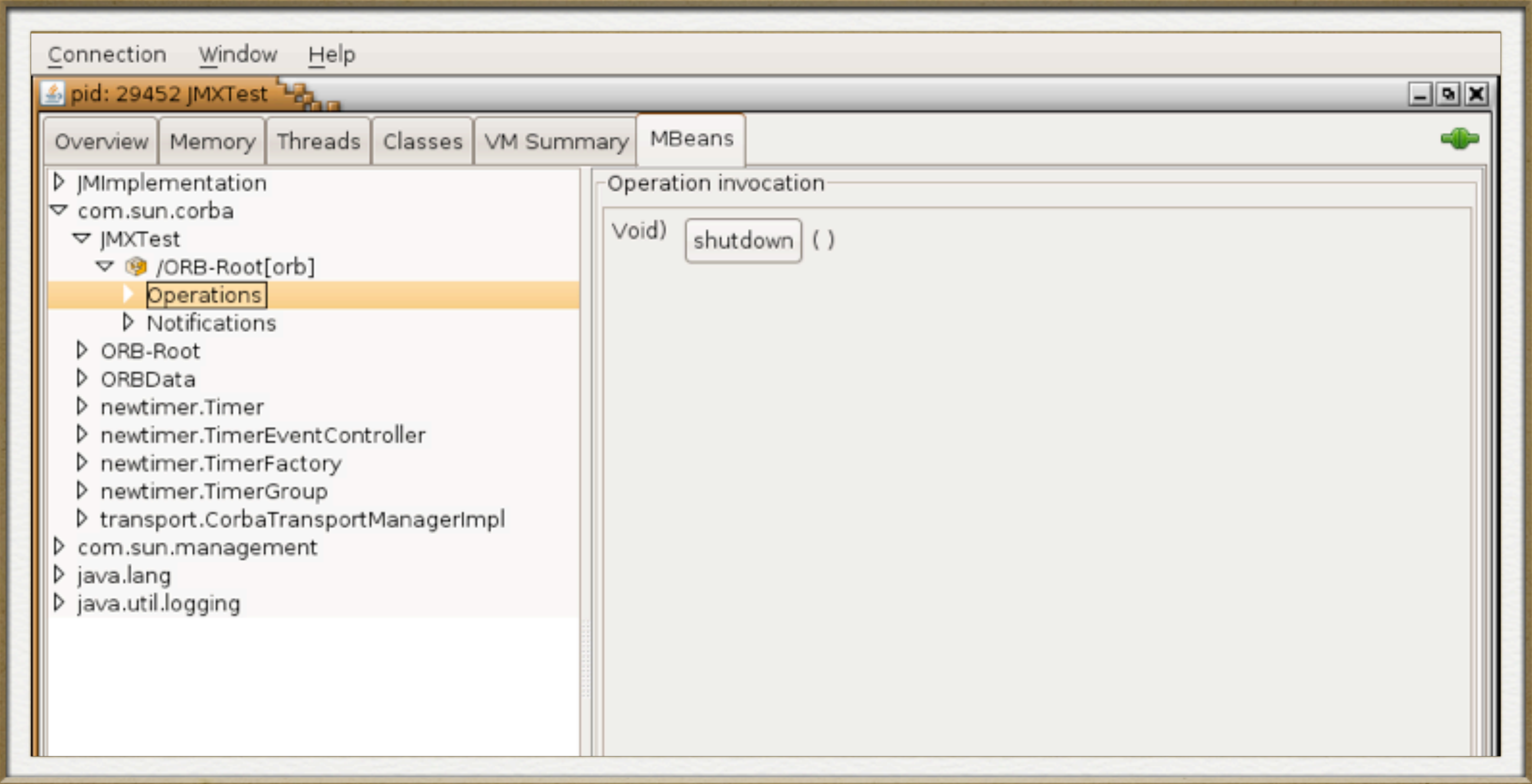
Username: **Password:**

Connecting to the test program

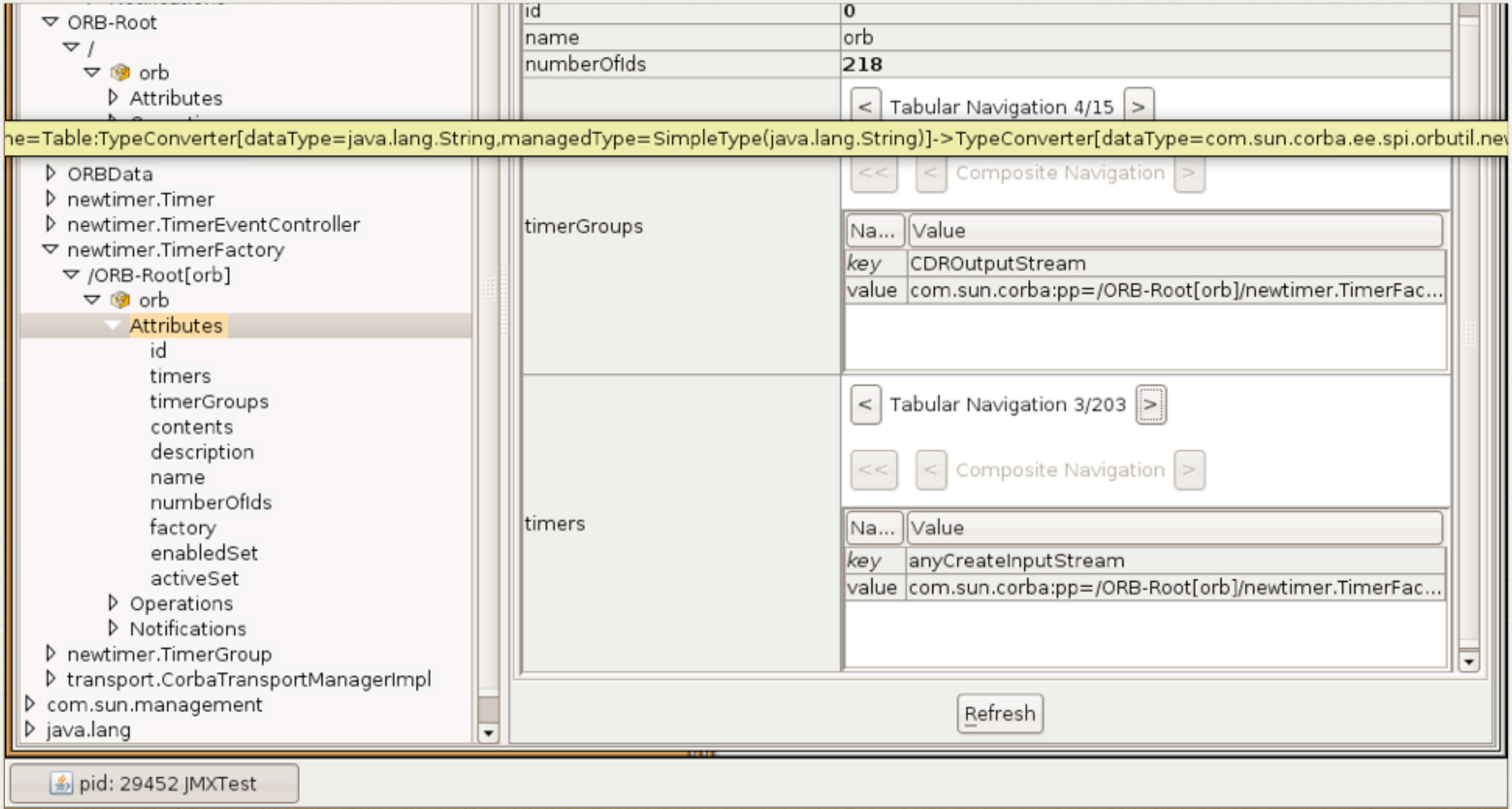


```
▷ JMImplementation
▽ com.sun.corba
    ▷ JMXTest
    ▷ ORB-Root
    ▷ ORBData
    ▷ newtimer.Timer
    ▷ newtimer.TimerEventController
    ▷ newtimer.TimerFactory
    ▷ newtimer.TimerGroup
    ▷ transport.CorbaTransportManagerImpl
▷ com.sun.management
▷ java.lang
▷ java.util.logging
```

Types of MBeans in Test



shutdown operation defined in test



ORB-Root
/
orb
Attributes

ne=Table:TypeConverter[dataType=java.lang.String,managedType=SimpleType(java.lang.String)]->TypeConverter[dataType=com.sun.corba.ee.spi.orbutil.ne

ORBData
newtimer.Timer
newtimer.TimerEventController
newtimer.TimerFactory
/ORB-Root[orb]
orb
Attributes
id
timers
timerGroups
contents
description
name
numberOfIds
factory
enabledSet
activeSet
Operations
Notifications
newtimer.TimerGroup
transport.CorbaTransportManagerImpl
com.sun.management
java.lang

id	0
name	orb
numberOfIds	**218**
timerGroups	< Tabular Navigation 4/15 > << < Composite Navigation >
timers	< Tabular Navigation 3/203 > << < Composite Navigation >

timerGroups:

Na...	Value
key	CDROutputStream
value	com.sun.corba:pp=/ORB-Root[orb]/newtimer.TimerFac...

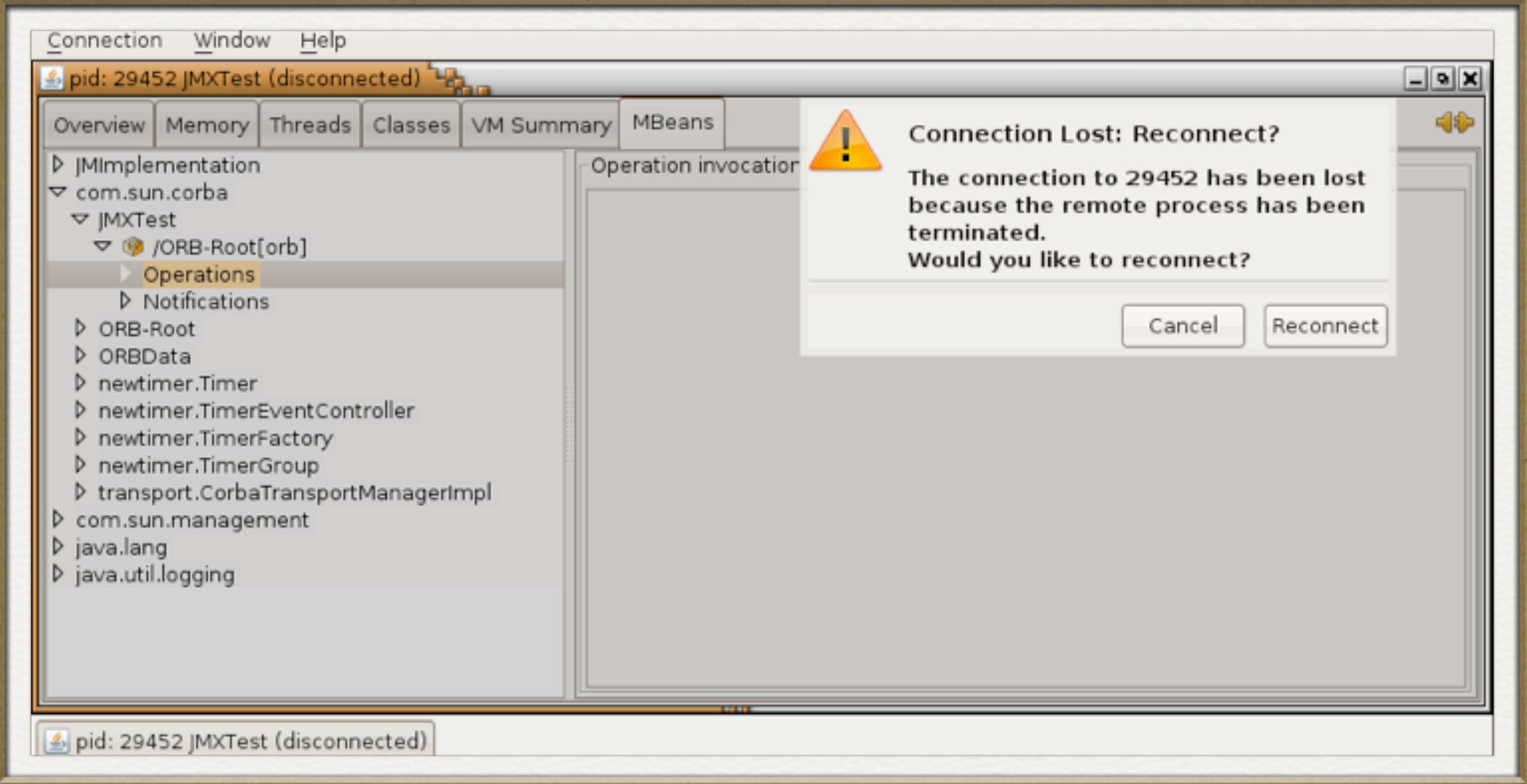
timers:

Na...	Value
key	anyCreateInputStream
value	com.sun.corba:pp=/ORB-Root[orb]/newtimer.TimerFac...

Refresh

pid: 29452 JMXTest

Attributes on TimerFactory



Connection Lost: Reconnect?

The connection to 29452 has been lost because the remote process has been terminated.
Would you like to reconnect?

Invoking the shutdown operation terminates the test process

Current Status

- Implementation Feature Complete
  - Some feature work from Metro and GFv3 Monitoring
  - Alignment with latest AMX is mostly done
- Most unit tests written
  - Some of the newer annotations need more testing
  - More testing needed on TypeConverter
- Working in ORB, Metro, and GF monitoring at least
- Will undoubtedly learn more as more people use it
- Currently 7 normal priority findbugs issues (2 in test, 2 in build time tools)

Workspace and Artifacts

- Project is at http://kenai.com/projects/gmbal
  - Currently master is the gmbal repository at http://kenai.com/hg/gmbal~master
- Gmbal also contains the monitoring probe client provider code
- Artifacts generated from gmbal project:
- Artifacts are all available in Maven

Group ID	Artifact ID	Current Version
org.glassfish.gmbal	gmbal-api-only	3.0.0-b002
org.glassfish.gmbal	gmbal	3.0.0-b002
org.glassfish.gmbal	gmbal-sources	3.0.0-b002
org.glassfish.provider	gfprobe-provider-client-source	3.0.0-b002
org.glassfish.provider	gfprobe-provider-client	3.0.0-b002

Code Coverage

package	class	method	block	line	Notes
gmbal	60%	19%	60%	26%	no-op impl not tested
generic	69%	38%	36%	28%	unused classes not tested
impl	66%	54%	40%	46%	AMXClient, TypeConverter need more testing
logex	100%	82%	86%	90%	adequately tested
typelib	95%	72%	59%	69%	simple classes: accept, hashcode, equals under-tested
util	100%	100%	74%	71%	A few error cases not covered

Open Issues

JIRA ID	Bug/RFE	Summary	Status
GMBAL-25	RFE	Add wildcard support to stripPrefix	Needs more testing
GMBAL-24	Bug	Regression in gmbal use in Metro	Appears to be JDK bug; fix in gmbal is to ignore the problem (field not used)
GMBAL-20	Bug	Need a better method to manage EvaluatedClassMap	Not started
GMBAL-18	RFE	Gmbal spends 30% of init time in constructing Exceptions implementation	Fix known; not clear if important enough to fix
GMBAL-17	RFE	Would like to ask if root name already registered	Still working on proper API for this
GMBAL-12	RFE	Autoencode Object names	Fix known: use ObjectName.quote
GMBAL-9	Bug	Need to add generation of description resource bundle for MBeans	Needed for I18N; fix known
GMBAL-8	RFE	Need to add resource bundle generation for gmbal exception logging	Needed for I18N; fix known
GMBAL-6	RFE	Align with MXBeans where possible	Understood but not started
GMBAL-5	RFE	Add benchmarks	Have one profile test for startup; more desired

Future Work

- Fully apply Gmbal to ORB
- Continue working with Gmbal users
- Possibly generate code at registration time using codegen for faster mbeans