Building and Configuring JXTA with Monitoring Capabilities

Version 1.0
5/21/03

Design Goals

Metering adds to both the processing and memory loads of the JXTA Stack. Similarly the metering code increases the size of the JXTA libraries. With this in mind:

The following requirements are respected in the MMP J2SE Binding architecture

- Since metering results in resource (memory, CPU) expenditures, it must be configurable
- It must be possible to build a JXTA Library without metering (to keep the minimal jxta.jar small)

To achieve this, the J2SE Binding will be designed to allow the following types of JXTA library builds:

- All Metering Off
  Metering unavailable: no metering memory/CPU load, smallest JAR)
  Metering capabilities are compiled out of the build
- Metering Always On
  Metering available: always active
  Metering capabilities are included in library build
- Metering Conditionally On
  Metering available: based upon setting of a user property
  Metering capabilities are included in library build
  This gives most flexibility, but with slight performance expense

- Service Metering Off (default if global metering is off)
  Service Metering unavailable: no metering memory/CPU load, smallest JAR)
  Service Metering capabilities are compiled out of the build
- Service Metering Always On (ignored if global metering is off)
  Service Metering available: always active
  Service Metering capabilities are included in library build
- Service Metering Conditionally On (ignored if global metering is off)
  Service Metering available: based upon setting of a user property
  Service Metering capabilities are included in library build
  This gives most flexibility, but with slight performance expense

Support Files for Configuring JXTA Builds

The Monitoring capabilities are determined by the state of two property files.

Property file used in the Build/Make/Ant process

File Name is specified in the ANT property:
net.jxta.build.meter.properties

Default value in the platform project's build.xml:
\platform\binding\java\build\conditionalBuild.properties

Property file used at JXTA runtime (in jxta.jar and/or in the classpath)
\net\jxta\user.properties

The **conditionalBuild.properties** file determines what monitoring functionality is compiled into the JXTA Stack as on/off/conditionally-on. The default values in the CVS tree produce a JXTA stack with all monitoring turned off.

The **user.properties** file determines is contains properties that can activate monitors that were specified as conditionally-on at build-time (ie via **conditionalBuild.properties**)

Configuration Files for Building JXTA with Metering Capabilities

The **conditionalBuild.properties** is used at build time to generate Java source files that are used at compile time to turn service monitors on/off/conditionally-on.

```
net.jxta.impl.meter.MeterBuildSettings.METERING=off, net.jxta.meter.conditionalMetering                    \

net.jxta.impl.endpoint.endpointMeter.EndpointMeterBuildSettings.ENDPOINT_METERING=off,                     \
                                                               net.jxta.meter.conditionalEndpointMetering

net.jxta.impl.endpoint.transportMeter.TransportMeterBuildSettings.TRANSPORT_METERING=off,                  \
                                                              net.jxta.meter.conditionalTransportMetering

net.jxta.impl.resolver.resolverMeter.ResolverMeterBuildSettings.RESOLVER_METERING=off,                     \
                                                               net.jxta.meter.conditionalResolverMetering

net.jxta.impl.rendezvous.rendezvousMeter.RendezvousMeterBuildSettings.RENDEZVOUS_METERING=off,  \
                                                             net.jxta.meter.conditionalRendezvousMetering
```

The properties in this file are of the form:

<package>.<className>.<static final boolean fieldName>

The values for these properties are a comma separated pair:

<on>|<off>|< runtime>
<variable for conditional metering used in user.properties>

If the **runtime** option is utilized, a one-time test will occur at load-time to determine whether to activate metering for this service based on an entry in the **user.properties** file

Note: The first property in this file **net.jxta.meter.MeterBuildSettings.METERING** is used as a base for all service Metering. If it is set to off, all other metering will be off, regardless of their settings

As part of the standard platform, there are three versions of the build properties file corresponding to metering always on, always off and metering determined by the user.properties file for each of the standard services:

build/meterOnBuild.properties
build/meterOffBuild.properties
build/meterRuntimeBuild.properties

Building JXTA with Metering Enabled

By Default, JXTA will be build with metering disabled. Namely:

```
D:\jxta\platform\binding\java> ant
```

Will result in the creation of dist/jxta.jar with metering disabled.

There are two ANT properties that can be used to build alternate metering:

net.jxta.build.meter.properties: Specifies the file to use when generating metering files

net.jxta.build.jxtaJar: Specifies the Jar file to create

So to build jxta.jar with metering on:

```
D:\jxta\platform\binding\java> ant -Dnet.jxta.build.meter.properties=build/meterOnBuild.properties
```

Alternatively you could create two jar files (and modify your classpath to determine whether metering is enabled):

```
D:\jxta\platform\binding\java> ant -Dnet.jxta.build.meter.properties=build/meterOffBuild.properties

D:\jxta\platform\binding\java> ant -Dnet.jxta.build.meter.properties=build/meterOnBuild.properties      \
                                        –Dnet.jxta.build.jxtaJar=dist/jxtaMetering.jar
```

This would result in the creation of two versions of JXTA:

dist\jxta.jar:	Standard JXTA with no metering capabilities
dist\jxtaMetering.jar:	Standard JXTA with full metering capabilities

Tuning Conditional Metering

The **user.properties** is used to activate/deactivate conditional/runtime metering.

```
net.jxta.impl.meter =on

net.jxta.impl.endpoint.endpointMeter=on

net.jxta.impl.endpoint.transportMeter=on

net.jxta.impl.resolver.resolverMeter=on

net.jxta.impl.rendezvous.rendezvousMeter=on
```

Valid values are **<on>** or **<off>**

Note: The first property in this file **net.jxta.impl.meter** is used as a base for all other conditional service Metering. If it is set to off, all other metering will be off, regardless of their settings.

Note: These properties are only utilized for metering options compiled with the **runtime** option specified.