Platform SDK Developer's Guide

Explicitly Choosing a Netty or Mina Connection Layer

12/16/2025

Explicitly Choosing a Netty or Mina Connection Layer

Problem: A user wants to explicitly specify which connection layer Platform SDK should use, instead of accepting the default one.

Resolution: Set the `com.genesyslab.platform.commons.connection.factory.class` system property (defined by the `ConnectionManager.CONN_FACTORY_KEY` constant) to the fully qualified name of connection factory class to use:

- `com.genesyslab.platform.commons.connection.impl.netty.NettyConnectionFactory` (or `NettyConnectionFactory.class.getName()`)
- `com.genesyslab.platform.commons.connection.impl.mina.MinaConnectionFactory` (or `MinaConnectionFactory.class.getName()`)

```
$> java -Dcom.genesyslab.platform.commons.connection.factory.class=
      com.genesyslab.platform.commons.connection.impl.netty.NettyConnectionFactory
      -jar your_application.jar
```

This property can be set either from command line using `-D` switch or from code, as shown in the sample below.

Please note that Platform SDK looks up connection factory each time a connection is created, allowing the user to use different connection layers for different connections.

```
package test;

import com.genesyslab.platform.commons.collections.KeyValueCollection;
import com.genesyslab.platform.commons.connection.Connection;
import com.genesyslab.platform.commons.connection.ConnectionManager;
import com.genesyslab.platform.commons.connection.configuration.KeyValueConfiguration;
import com.genesyslab.platform.commons.connection.impl.netty.NettyConnectionFactory;
import com.genesyslab.platform.commons.connection.impl.mina.MinaConnectionFactory;
import com.genesyslab.platform.commons.protocol.Endpoint;
import com.genesyslab.platform.commons.protocol.ServerChannel;
import com.genesyslab.platform.management.protocol.solutioncontrolserver.runtime.channel.
   SolutionControlServerInternalProtocolFactory;

public class TlsServer {
  private ServerChannel channel = null;

  public void start() throws ProtocolException, InterruptedException {
    // System.setProperty(ConnectionManager.CONN_FACTORY_KEY,
    // NettyConnectionFactory.class.getName());
    System.setProperty(ConnectionManager.CONN_FACTORY_KEY,
    MinaConnectionFactory.class.getName());
    KeyValueCollection kvc = new KeyValueCollection();
    kvc.addString(Connection.TLS_KEY, "1"); // Turn on TLS
    // Java keystore format, use "keytool" utility from JDK to generate one
    kvc.addString(Connection.SSL_KEYSTORE_PATH_KEY, "D:\\Test\\keystore");
    kvc.addString(Connection.SSL_KEYSTORE_PASS, "password");
    Endpoint endpoint = new Endpoint("TlsServer", "localhost", 5500);
    channel = new ServerChannel(endpoint, new SolutionControlServerInternalProtocolFactory());
```

```
    channel.configure(new KeyValueConfiguration(kvc));
    channel.open();
  }
  // And so on...
}
```