Java News

synchronized and static

```
/**
 * Insert the method's description here. Creation date: (8/24/01 3:51:37 PM)
 *
 * @return de.desy.tine.client.TLinkFactory
 */
public synchronized static TLinkFactory getInstance()
{
   if (instance == null) instance = new TLinkFactory();
   return instance;
}
/**
 * Put a new TLink in the link table Creation date: (8/17/01 5:59:27 PM)
 *
```

Unsynchronized and static

```java
public class TLinkFactory
{
  private static TLinkFactory instance = new TLinkFactory();
  boolean active;
  */
  private TLinkFactory()
  {
    initFactory();
  }
```

Class loaded makes the first instance !

Constructor is private !

```java
  /**
   * Insert the method's description here. Creation date: (8/24/01 3:51:37 PM)
   *
   * @return de.desy.tine.client.TLinkFactory
   */
  public static TLinkFactory getInstance()
  {
    if (instance == null) instance = new TLinkFactory();
    return instance;
  }
```

Get rid of the 'synchronized' !

Don't need !

Other Singletons in tine.jar

- TEquipmentModuleFactory
- TInitializerFactory

Multicast Sockets and java

- Point: You don't have to join a multicast group to send a multicast ! (always true).
- Java Catch-22:
  - Need a "Time To Live" > 1 in order to pass by the first router
  - Can't set the Time To Live in java unless the socket is declared as a MulticastSocket (not just a DatagramSocket)!
  - MulticastSockets in java AUTOMATICALLY set the "reuse address" option ! (Not Good for TINE sockets in general!)
  - Release 4.0: needs special sockets to send multicasts !

Depends on the Constructor !

```
}
else if (port == initializer.getMCastPort())
{
  grp = InetAddress.getByName(initializer.getMCastAddress());
  m = new MulticastSocket(port);
  m.joinGroup(grp);
  m.setSoTimeout(0);
  m.setReceiveBufferSize(0x10000);
  m.setTimeToLive(16);
  socketBound = true;
}
else if (port == initializer.getNetCastPort())
{
  grp = InetAddress.getByName(initializer.getNetCastAddress());
  m = new MulticastSocket(port);
  m.joinGroup(grp);
  m.setSoTimeout(0);
  m.setReceiveBufferSize(0x10000);
  socketBound = true;
}
```

A Way Around the Catch-22

```java
int clnport = initializer.getClnPort();
boolean useNextPort = false;
boolean socketBound = false;
InetAddress grp;
for (int i=0; !socketBound && i<1000; i++)
{
  try
  {
    if (port == 0)
    {
      useNextPort = true;
      //s = new DatagramSocket(clnport+i);
      InetAddress localAddr = InetAddress.getLocalHost();
      InetSocketAddress addr = new InetSocketAddress(localAddr, clnport+i);
      s = new MulticastSocket(null);
      s.setReuseAddress(false);
      s.bind(addr);
      s.setSoTimeout(0);
      s.setReceiveBufferSize(0x10000);
      s.setTimeToLive(16);
      socketBound = true;
    }
```

New 'feature' (Release 3 and 4)

- **Problem:**
  - Server has property "Energy" as FLTINT (a value/status pair).
  - Caller want to ignore the status part and only get the Energy as a float.
- **C,C++ server:**
  - Just overload the property! Trap on the requested data type and give the caller what he wants!
- **VB:**
  - Automatically overloaded (as far as possible)
- **Java (TINE interface)**
  - Just overload the property! (Add a property handler for each supported data type).
- **Java (Device Server Wizard)**
  - Oops!

- Allow systematic format conversion from 'doublets' to 'singlets' for READ calls!
  - FLTINT -> float; INTINT -> 1st int; DBLDBL -> 1st DBL
- NOTE: format conversion among primitives already available
  - e.g. registered float can be obtained as a double, etc.

Release 4.0 News

- TQuery now queries for longer names!
  - Needed for Central Archive and Central Alarm Servers
- New TAlarmSystem Class
  - analogous to THistory Class
- XML configuration files logic now in place.
- Complex Tagged Structures underway
- Release 4.0 protocol to come next.