HTest v0.3

Java JPOS ISO8583 testing tool

HTest, is a java jpos based ISO8583 tool for testing and development purposes.

It's basically a web front end for jpos ee q2 services that I created and use for my iso8583 testing.

Htest have the features that made it easier for me to test, debug and develop ISO8583 payment systems

Familiarity with JPOS library and framework is recommended to use this tool but not required.

It provides the following features :

- o Create and edit ISO8583 server and client connections via web based gui

- o Beanshell scriptable ISOchannel that provides raw hexdump of channel messages.

- o Auto conversion from ISO message to http xml format for easier webservice integration

To run HTest you need java 1.6 installed(I think, I can't remember wether I used 1.3 compatible api)

It works on any Operating Systems that have java support.

Download and unzip Htest.zip

Go to the unziiped folder then type "*java –jar Htest.jar*" or "java -jar jpos-ee.jar"

After that point your browser to http://localhost:9090/jposee

Login with username : admin : password : test
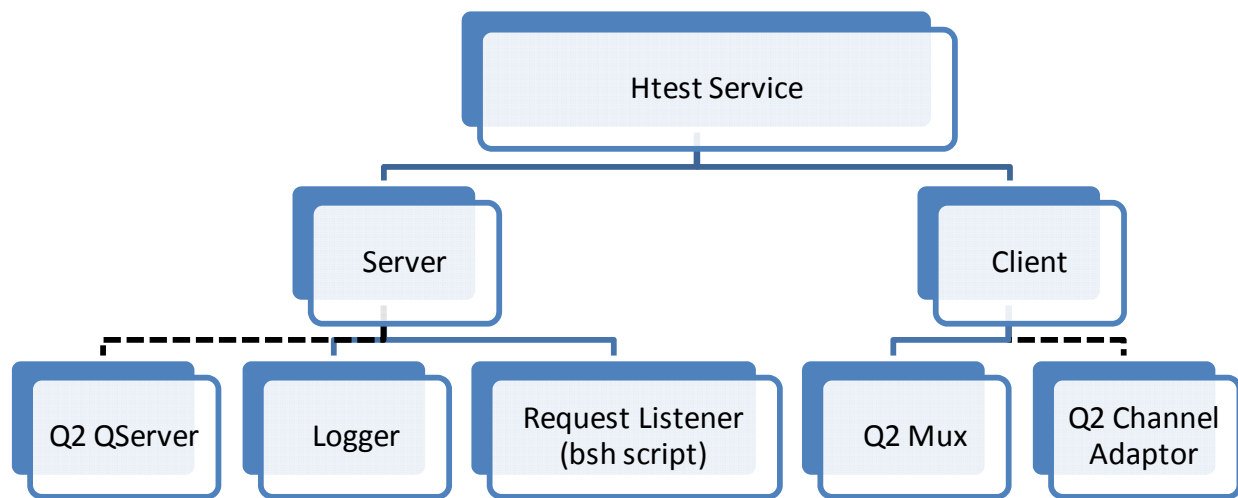
File Structure

In the root folder

All the configuration file is in cfg/isotest

All the jpos q2 deployment descriptor is in deploy/

Copyright Hairi Abass 2011, HTest v0.3 a Web GUI for JPOS Q2

General Concepts

In HTest we define the term services as a representation to either JPOS QServer or ChannelAdaptor.

Which is basically either an ISO8583 listening server or a client.

When you create a HTest Service it is actually an abstract representation of several xml files that JPOS Q2 utilize. All the xml Q2 config when created is stored in the cfg/isotest/servicedef folder

If the user clicks start, the files are simply copied to the deploy directory and the JPOS Q2 container will initialize them.

The request listener script with .bsh extention handles what do to when an ISO message arrive at the service. The requestlistener script is located at cfg/isotest/servicedef/support
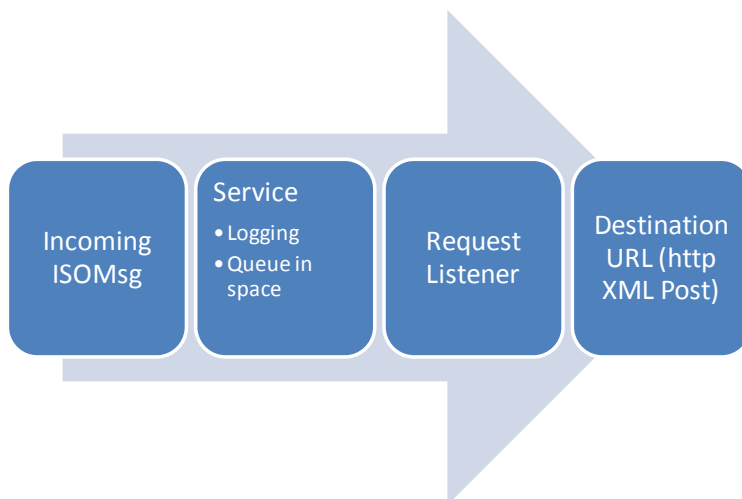
1. **Creating Services :**

Services in Htest means the Server and Client ISO8583 connection and the request listener that handles the incoming request.

You can easily create and modify services in Htest by using the ISO8583 Test Services→Create Service Menu

To Create a service just enter the desired servicename,

And fill out the needed details, such as port number, ISOChannel type and packager.

Optionally you can also include a http url to post the incoming ISOmessage from that service.

There are two service type the server and client service. If you want to create a client service you also need to specify the destination host/ip address.

For channel type use the custom channel option if your creating / debugging a new ISOchannel that is not in the standard Jpos library. It is a bsh scriptable channel with HEXDUMP output in the log, helps you prototype and debug errors more easily. To create a new custom channel do it in the configurations menu.

Copyright Hairi Abass 2011, HTest v0.3 a Web GUI for JPOS Q2

2. Monitor Services

Access via ISO8583 Test Services → List Services

Displays all the currently available services and their info

HTest

Welcome admin

Java JPOS Web based ISO8583 Testing Tools

ISO8583 Test Service   Configurations

Message Box :	

ServiceName	ServiceType	Data Type	Service Address	START/STOP	EDIT	ACTION	INFO
as	Server	ASCIIChannel base1	: 3456	START	[VIEW]	[SEND MESSAGE]	
mysss	Server	channel1 iso87ascii	: 10030	START	[VIEW]	[SEND MESSAGE]	
ss	Client	ASCIIChannel base1	localhost : 3456	START	[VIEW]	[SEND MESSAGE]	
sss	Server	ASCIIChannel base1	: 0	START	[VIEW]	[SEND MESSAGE]	
testclean2	Server	ASCIIChannel iso87ascii	: 10030	START	[VIEW]	[SEND MESSAGE]	

Use the start/stop to start or stop a service,

If service is active all their connection related info is displayed in the info tab.

To send an ISOMessage through that service click the send Message link

3. **Send Message**

Access this menu via ISO8583 Test Service → List Service then click on the link Send MessageTool for
Sending ISOMessages trough services and seing the log files.

ISO Message is inputted to the text field in the JPOS XML format like this

<isomsg direction="incoming">
<field id="0" value="0810"/>
 <!-- Example Field, Fill it with whatever you want your isomesage to contain -->
<field id="7" value="12345"/>
</isomsg>

The resulting output is then displayed in the log output.
The logoutput will contain info as the time of message send, outgoing and incoming message, plus the
exception generated if any.

The count variable sets the number of time you wish to send the message with delay is the delay
interval between each message in ms.

4. Interfacing

HTest was written to support primarily ISO8583 to webservice/http xml style connection.

So if you filled the incomingURL field when you were creating the ISO8583 service, every incoming ISOMsg is posted to that url with HTTP POST method and fieldname isomessage.

Now to send an ISO8583 message to a service, besides using the JavaSpace interface, HTest provides the following http POST url
http://yourhost:9090/jposee/isotest/send_post.html?isomessage=xxxxx&servicename=xxxxx
Where servicename contains the name of your service.

If it is a client service, your iso message will be sent to connected server.
If it is a server service, your iso message will be sent to the **LAST CONNECTED CLIENT**

If you want to chain several ISO8583 services together heres what you do.

Say you have Service A and Service B

On the incomingURL parameter of ServiceA you put

http://yourhost:9090/jposee/isotest/send_post.html?servicename=ServiceB

On the incomingURL parameter of ServiceB you also put

http://yourhost:9090/jposee/isotest/send_post.html?servicename=ServiceA

There, you have it any incoming msg to Service A will be passed to Service B and vice Versa

And of course should you decide to change all of this the beanshell script is available for perousing at Cfg/isotest/servicedef/support/servicename.bsh
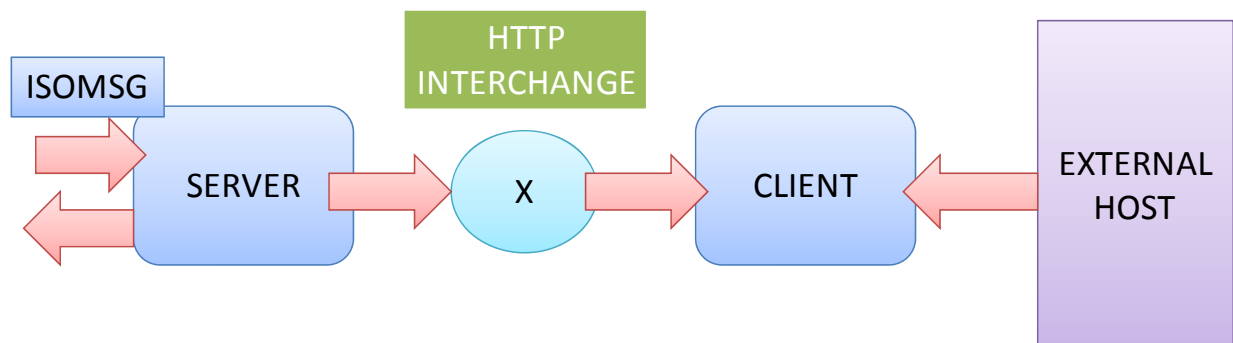
5. Configurations

Now on the configurations menu you can create and edit the custom channels, predefined iso message and packager xml config.

For ISOMsg configurations its just there to let you create and save your iso message to save time during your testing.

The channel configuration is for the custom channels, you can customize the sendMessageLength, getMessageLength, sendHeader and getHeader.
These four function should be enough for most tcp based ISO8583 connection.

The packager configurations, is there for you to customize your ISO8583 field definitions, wether its LLChar, LLNum , field length and et cetera.

6. Conclusion

Well that's pretty much it, Htest is a tool I've developed to help me speed up my testing, since I usually connect to many different terminals with slightly different channels and message type.
I hope it can be of use to all of you doing payment system development.

Contact me at
hairi@m-sinergi.com