Click on broker.bat to start the broker
Example 1: Screen Shots

**NB SimpleClient 1**

- configuration: load D:\LAB-July2004\MaradaBrokering-0.97\config\ServiceConfiguration.txt
- Duration: 8.6 50000 40000
- FragmentationService: Fragmentation Directory - D:/TempFiles/tmpFiles/fragment
- CoalescingService: Coalescing Directory value = D:/TempFiles/tmpFiles/coalesce
- SessionService: Initializing services using file D:\LAB-July2004\MaradaBrokering-0.97\config\ServiceConfiguration.txt
- NIOTCPLinkFactory: Property NIOTCPServerPort not specified. So this is a client.
- NIOTCPLinkFactory: Elapsed time for establishing connection = 0 mSecs
- TransportHandlerImpl: Registering link [localhost:3045]
- CommunicationsService: Client registration success!

Total offset computed at initialization: 9532

**NB SimpleClient 2**

- configuration: load D:\LAB-July2004\MaradaBrokering-0.97\config\ServiceConfiguration.txt
- Duration: 8.6 50000 40000
- FragmentationService: Fragmentation Directory - D:/TempFiles/tmpFiles/fragment
- CoalescingService: Coalescing Directory value = D:/TempFiles/tmpFiles/coalesce
- SessionService: Initializing services using file D:\LAB-July2004\MaradaBrokering-0.97\config\ServiceConfiguration.txt
- NIOTCPLinkFactory: Property NIOTCPServerPort not specified. So this is a client.
- NIOTCPLinkFactory: Will not be starting the ServerSocket Thread since this is a client.
- NIOTCPLinkFactory: Elapsed time for establishing connection = 0 mSecs
- TransportHandlerImpl: Registering link [localhost:3045]
- CommunicationsService: Client registration success!

Total offset computed at initialization: 9535
NB SimpleClient 1

- configuration: load D:\LAB-July2004\NaradaBrokering-0.97\config\ServiceConfiguration.txt
- duration: 0.8 50000 40000
- FragmentationService: Fragmentation Directory value - D:\TempFiles/tmpFiles/fragment

CoalescingService: Coalescing Directory value - D:\TempFiles/tmpFiles/coalesce
SessionService: Initializing services using file 'D:\LAB-July2004\NaradaBrokering-0.97\config\ServiceConfiguration.txt'
NIOICPLinkFactory: Property NIOICPServerPort not specified. So this is a client.
NIOICPLinkFactory: Will not be starting the ServerSocket Thread since this is a client.
NIOICPLinkFactory: Elapsed time for establishing connection =
NIOICPLinkFactory: Time to establish connection to [localhost:3045] = 0 mSec
TransportHandlerImpl: Registering link [localhost:3045]
CommunicationsService: Client registration successful.
OutputStream::Casahlanca
Total offset computed at initialization: 9532
PrintStreamImpl: hello there?

NB SimpleClient 2

- configuration: load D:\LAB-July2004\NaradaBrokering-0.97\config\ServiceConfiguration.txt
- duration: 0.8 50000 40000
- FragmentationService: Fragmentation Directory value - D:\TempFiles/tmpFiles/fragment

CoalescingService: Coalescing Directory value - D:\TempFiles/tmpFiles/coalesce
SessionService: Initializing services using file 'D:\LAB-July2004\NaradaBrokering-0.97\config\ServiceConfiguration.txt'
NIOICPLinkFactory: Property NIOICPServerPort not specified. So this is a client.
NIOICPLinkFactory: Will not be starting the ServerSocket Thread since this is a client.
NIOICPLinkFactory: Elapsed time for establishing connection =
NIOICPLinkFactory: Time to establish connection to [localhost:3045] = 0 mSec
TransportHandlerImpl: Registering link [localhost:3045]
CommunicationsService: Client registration successful.
OutputStream::Casahlanca
Total offset computed at initialization: 9535
SimpleClient: Received NBEvent hello there? [1]
Example 2: Screen Shots (Compression and Decompression)
NB CompressionClient1

- configuration: load P:\nb0972\config\ServiceConfiguration.txt
duration: 0.0 50000 40000
FragmentationService: Fragmentation Directory value - C:/TempFiles/tmpFiles/Frag
cent
CoalescingService: Coalescing Directory value - C:/TempFiles/tmpFiles/coalesce
SessionService: Initializing services using file P:\nb0972\config\ServiceConfig
uration.txt
NIOTCPLinkFactory: Property NIOTCPServerPort not specified. So this is a client
NIOTCPLinkFactory: Will not be starting the ServerSocket thread since this is a client
NIOTCPLinkFactory: Elapsed time for establishing connection =
NIOTCPLinkFactory: Time to establish connection to [localhost:3045] = 0 mSec
TransportHandlerImpl: Registering link [localhost:3045]
CommunicationsService: Client registration success!
Movie/Casablanca
Total offset computed at initialization: -1400
CompressionClient: Received Decompressed NBEvent with payload >>>>>>>>>>>>>>>>
{hello} message num 1(

NB CompressionClient2

- configuration: load P:\nb0972\config\ServiceConfiguration.txt
duration: 0.0 50000 40000
FragmentationService: Fragmentation Directory value - C:/TempFiles/tmpFiles/Frag
cent
CoalescingService: Coalescing Directory value - C:/TempFiles/tmpFiles/coalesce
SessionService: Initializing services using file P:\nb0972\config\ServiceConfig
uration.txt
NIOTCPLinkFactory: Property NIOTCPServerPort not specified. So this is a client
NIOTCPLinkFactory: Will not be starting the ServerSocket thread since this is a client
NIOTCPLinkFactory: Elapsed time for establishing connection =
NIOTCPLinkFactory: Time to establish connection to [localhost:3045] = 0 mSec
TransportHandlerImpl: Registering link [localhost:3045]
CommunicationsService: Client registration success!
.Movie/Casablanca
Total offset computed at initialization: -1400
Payload length=41
compressedDataLength=36
CompressionClient: Received Decompressed NBEvent with payload >>>>>>>>>>>>>>>>
{hello} message num 1(}
Example 3: Reliable delivery of events

Step 1: Double click `broker.bat`. This starts the broker.

Step 2: Next, double click `robustNode.bat`. This starts robust delivery service and the underlying stable storage. The GUI for this application is depicted below.

Specify the broker-IP **address** and the **Entity Id** (it could be anything e.g. 888) for the robust node. Proceed to press the **Start** button. If all goes well in the connection to the broker and the setup of the JDBC connections you will see the GUI depicted in Figure 2.
Figure 2: The GUI after the successful setup of the broker connection and the storage service
Proceed to click the **View Table** tab. If this is the first time that you are running this application everything will be empty and there will be no entries in the sub tabs that are available e.g. **Inventory**, **Template**, **Profile** and **EntityTemplate**.

![View Table Tab](image)

**Figure 3: Viewing previously registered templates**
The figure below depicts the scenario where there are no entities registered to any templates. In fact it is also possible (as can be seen by clicking the Template tab) that there are no registered templates. If this is the case proceed to add entries regarding the template, entity and register an entity to the template in question. To add information regarding the entities, templates etc. click on the Edit Table tab. The screen that is displayed is depicted below.

![Figure 4: Viewing the templates and entities registered to these templates.](image)
The screen that comes up when you click on the **Edit Table** tab is depicted below. First you need to add the right template in question. Proceed to click on the **Edit Template** tab.

![Image](image.png)

**System response:**

![System Response](image.png)

**Figure 5:** The Edit Table screen with sub tabs for registrations
The screen that is displayed when you click the **Edit Template** tab is depicted below. The default that is displayed is Movies/Casablanca with a **templateId** of 11111. Note that for the GridFTP demo you will have to remove the s from Movies and have the string as Movie/Casablanca if you are setting up the GridFTP from scratch.

![Image of registering a new template - Initial screen](image)

Figure 6: Registering a new template - Initial screen
Next proceed to click the **Edit Entity** Tab. The screen that is now displayed is depicted below. The display depicts the entries currently available. We are interested in adding entities with ID **4444** and **5555**. If you don’t see the entries in the list proceed to register the entities. Note that this is a one-time operation and you will not need to do this again until the database/file-system is cleared.

**Figure 7: The screen for editing entity entries.**
The figure below depicts the process of adding entities 4444 and 5555. Next we need to register these entities to the templateId in question. For this you need to click on the Entity EntityTemplate tab.

Figure 8: The screen after adding entries for entity 4444 and 5555
Next, proceed to register entities to a template by clicking on the **Edit Entity Template** tab. The screen below shows the registering of an entry to a template. The highlighted items indicate the entity and templateId for which registration/deregistration would be performed when the appropriate buttons are pressed.

The figure below depicts entity **4444** being registered to templateID **11111**

![Figure 9: The screen for registering an entity to a template](image-url)
The figure below depicts entity 5555 being registered to templateID 11111.

**Figure 10:** Another example depicting the registration of an entity to a template.
Note that the registration process involving entities, templates and entities-to-templates is a one time operation for the life of the application. You need not repeat the set of instructions detailed earlier if you are returning from a scheduled downtime or a failure. The operations need to be repeated ONLY if you have explicitly deregistered the entity, template or entities-from-the-template.

The figure below shows the screen, which allows one to see the templates that were registered.

![Figure 11: Viewing the registered templates](image)

Figure 11: Viewing the registered templates
The figure below depicts the entity and the templates that they are registered to.

Figure 12: Viewing entities and the templates that they are registered to
Now we proceed to running the robust subscriber. When you click on the `RobustSub.bat` file this is the GUI through which your actions are initiated. In the connection screen you need to enter the hostname and port number information.

![Figure 13: The robust subscriber GUI -- Connect Tab](image)

The figure below depicts the recovery screen. You need to specify the `templateId (11111)` on which you are going to recover. If you had previously subscribed, you will automatically be registered to those subscriptions. However, since the first time you need to specify your subscription.

![Figure 14: The robust subscriber recovery screen](image)
The subscription profile screen is depicted below. Here you need to specify both the templateId (11111) and the subscription (Movies/Casablanca).

![Subscription Screen](image)

Figure 15: Robust Subscriber - Subscription screen

The figure below depicts the scenario where, if you had previously registered a subscription, you will automatically be registered to that subscription. This obviates the need to subscribe using the Profile screen.

![Recovery Screen](image)

Figure 16: Robust Subscriber - Recovery screen depicting automatic subscriptions
The publisher publishing screen is similar to what’s available for the robust subscriber.

![Diagram of RobustPublisher - Connection Screen](image17)

Figure 17: RobustPublisher - Connection Screen

Specify the **templateId** (11111) and the **template** (Movies/Casablanca) of the events that the robust publisher will publish.

![Diagram of RobustPublisher - TemplateInfo screen](image18)

Figure 18: RobustPublisher - TemplateInfo screen
The figure below depicts the scenario where a publisher publishes a message and the robust subscriber receives it.

Figure 19: Publisher sending events and receiver receiving them
Example 4: Replay Client
Make sure that the subscriber in the previous case has subscribed and retrieved events. You can also click the View Table -> Inventory tab of the RDS node to confirm that you do have sequences that match your replay query. Here you can see that there are 3 events with sequence numbers 1, 2, 3 that are for templateId 11111. Make sure that the subscriber in the earlier demo had subscribed to events.
fragmentation: 0.4
duration: 0.8
50000 40000
FragmentationService: Fragmentation Directory value - D:/TempFiles/tmpFiles/fragment
CoalescingService: Coalescing Directory value - D:/TempFiles/tmpFiles/coalesce
SessionService: Initializing services using file D:\LAB-July2004\NaradaBrokerinner\btw\config\ServiceConfiguration.txt
NIOICPLinkFactory: Property NIOICPServerPort not specified. So this is a client
NIOICPLinkFactory: Will not be starting the ServerSocket Thread since this is a client
NIOICPLinkFactory: Elapsed time for establishing connection = 0 mSecs
TransportHandleImpl: Registering link [localhost:3045]
CommunicationsService: Client registration success!

Total offset computed at initialization: 9515

Usage:
r <start> <end> ---r start, end indicate range of values to be replayed

num 3
i 3
4444/ReplayServices
[i1] [21] [3]

ReplayClient: TemplateId=11111, EntityId=4444, ReplayIdentifier=9ed5c9b8-2894-4c08-bd5f-ae271f6d5a7e, Success: Additional Info=I have zero (0) that were considered for replay, Since no additional constraints have been specified, the replay service will route all the sequences
There are (3) sequences to replay ->

ReplayClient: hey there

ReplayClient: hello

ReplayClient: how are you
Example 5: Coalescing and Fragmentation Services:
The following screenshots depict the process of creating the right templateId, topic and entities to make the examples work.
TemplateId=54321 Topic=Sports/NBA
EntityIds=2222 and 3333

<table>
<thead>
<tr>
<th>Template Type :</th>
<th>String</th>
<th>XML</th>
<th>Integer</th>
<th>Regular Expression</th>
<th>Tag Value Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template Id :</td>
<td>54321</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sports/NBA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

System response:
Input Register Entity: 3333

4444
5555
2222
3333

Select Deregister Entity:

System response:
<table>
<thead>
<tr>
<th>Select Entity ID</th>
<th>Select Template ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>4444</td>
<td>11111</td>
</tr>
<tr>
<td>5555</td>
<td>54321</td>
</tr>
<tr>
<td>2222</td>
<td></td>
</tr>
<tr>
<td>3333</td>
<td></td>
</tr>
</tbody>
</table>

System response:
<table>
<thead>
<tr>
<th>Select Entity ID</th>
<th>Select Template ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>4444</td>
<td>11111</td>
</tr>
<tr>
<td>5555</td>
<td>54321</td>
</tr>
<tr>
<td>2222</td>
<td></td>
</tr>
<tr>
<td>3333</td>
<td></td>
</tr>
</tbody>
</table>

**System response:**

Profile update request was not successful
   ->ReliableDeliveryServiceImpl: Profile [StringProfile - [Sports/NBA] Destinations = [3333Id = [Sports/NBA]>3333]] was previously registered!
SubscriberPercolator: Received recovery response with SubscriberInfo
Invocations: There were no missing sequences on template [54321]. Recovery process is thus complete!
NdEntityActuatorImpl: ->EventConsumerImpl: Profile is already registered StringProfile = [Sports/NBA] Destinations = [3333Id = [Sports/NBA]>3333]

CoalescingClient: NBRecoveryNotification:
   EntityId=3333, TemplateId=54321, RecoveryId=1090346748390 SUCCESS=true
   The following profiles were PREVIOUSLY registered
StringProfile = [Sports/NBA] Destinations = [3333Id = [Sports/NBA]>3333]
   Entity has been automatically subscribed to all previously registered profiles

Total offset computed at initialization: -1402

NIOConnectionFactory: Time to establish connection to [localhost:3045] - 0 mSecs
TransportHandlerImpl: Registering link [localhost:3045]
CommunicationService: Client registration success!
   2222/Negotiations
2222/Recovery/Publisher
FragmentationClient: Assigned recovery id = [1090346865843]

FragmentationClient.main() ->Type h for usage options

FragmentationService: C:\TempFiles\tmpFiles\fragment is NOT a directory.
FragmentationClient: NBRecoveryNotification:
   EntityId=2222, TemplateId=54321, RecoveryId=1090346865843 SUCCESS=true
   Recovery process updated the publishers extension number to 101

Total offset computed at initialization: -1404

h
Usage: One of two possible nodes
   num <fileName> <numOfFragments> ----> Fragment based on the num of fragments
   size <fileName> <fragmentSize> ----> Fragment based on the size of fragments
num sqlManual.html 10
FragmentationService: The specified file [sqlManual.html] is of length - 3167400
FragmentationService: Beginning fragmentation process!
FragmentationService: Will create and store fragments to [C:\TempFiles/tmpFiles/
Fragment/sqlManual.html-tempDir]
FragmentationService: Total bytes written - 3167400
Original File Location: sqlManual.html
Total Number of Fragments: 10
Fragment Size: 316740
Last Fragment Size: 316740
Event's Content Synopsis Sports/NBA
FragmentationService: Fragmentation process complete..
FragmentationService: FragmentInfo file [C:\TempFiles/tmpFiles\fragment\sqlManua
1.html-tempDir\Fragment1.nbl] deleted!
FragmentationService: FragmentInfo file [C:\TempFiles/tmpFiles\fragment\sqlManua
1.html-tempDir\Fragment2.nbl] deleted!
FragmentationService: FragmentInfo file [C:\TempFiles/tmpFiles\fragment\sqlManua
1.html-tempDir\Fragment3.nbl] deleted!
FragmentationService: FragmentInfo file [C:\TempFiles/tmpFiles\fragment\sqlManua
1.html-tempDir\Fragment4.nbl] deleted!
FragmentationService: FragmentInfo file [C:\TempFiles/tmpFiles\fragment\sqlManua
1.html-tempDir\Fragment5.nbl] deleted!
FragmentationService: FragmentInfo file [C:\TempFiles/tmpFiles\fragment\sqlManua
1.html-tempDir\Fragment6.nbl] deleted!
FragmentationService: FragmentInfo file [C:\TempFiles/tmpFiles\fragment\sqlManua
1.html-tempDir\Fragment7.nbl] deleted!
FragmentationService: FragmentInfo file [C:\TempFiles/tmpFiles\fragment\sqlManua
1.html-tempDir\Fragment8.nbl] deleted!
FragmentationService: FragmentInfo file [C:\TempFiles/tmpFiles\fragment\sqlManua
1.html-tempDir\Fragment9.nbl] deleted!
FragmentationService: FragmentInfo file [C:\TempFiles/tmpFiles\fragment\sqlManua
1.html-tempDir\Fragment10.nbl] deleted!
CoalescingService: Processing fragment 10 of file sqlManual.html
CoalescingService: Wrote fragment [fragment10.nhf] of size = 3167480
CoalescingService: Fragments are available for coalescing The coalesced file is of size = 3167480

CoalescingService: Will delete 10 fragments
CoalescingService: Deleting [fragment1.nhf]
CoalescingService: Deleting [fragment2.nhf]
CoalescingService: Deleting [fragment3.nhf]
CoalescingService: Deleting [fragment4.nhf]
CoalescingService: Deleting [fragment5.nhf]
CoalescingService: Deleting [fragment6.nhf]
CoalescingService: Deleting [fragment7.nhf]
CoalescingService: Deleting [fragment8.nhf]
CoalescingService: Deleting [fragment9.nhf]
CoalescingService: Deleting [fragment10.nhf]

CoalescingService: Created coalesced file -> C:\TempFiles\tmpFiles\coalesce\sqlManual.html

CoalescingClient: Received COALESCED file IC:\TempFiles\tmpFiles\coalesce\sqlManual.html of length=3167480
CoalescingService: Total Memory Utilization is 5824512 bytes
Example 6: JMS application
Type h for usage options
NJOPClientFactory: Property NJOPServerPort not specified. So this is a client
NJOPClientFactory: Will not be starting the ServerSocket Thread since this is a client
NJOPClientFactory: Elapsed time for establishing connection = 0 mSec
TransportHandlerImpl: Registering link [localhost:3045]
JmsTopicConnectionFactory: Waiting for initialization of the JMS bridge.
Initialization of the bridge [Id:10983476466] complete after (28) milliseconds
Will be publishing to topic [/streams/chat/private/text]
Subscribing to [/streams/chat/private/text]
[john]: Hello there (1)
hi
[tony]: hi (1)