OMD-C ON-BOARDING TOOLS
USER GUIDE

HKEX Orion Market Data Platform
Securities Market & Index Datafeed Products

Version 2.6
16 July 2020

DOCUMENT HISTORY

Distribution Version

Version	Date of Issue	Comments
V1.0	20 July 2012	First Distribution Issue
V1.1	18 October 2012	Updated to include multi-machine config and usage of two sets of market data.
V1.1.1	31 October 2012	Correction in command line options for XDP-Dump(3.2). Multicast addresses in Appendix updated.
V1.1.2	25 June 2013	Appendix B – FAQ and Troubleshooting
V1.2	2 August 2013	Updated to include derivatives market products Reorganize the 3 Appendixes
V1.3	22 November 2013	Updated the canned data with the security market production data Updated the OMD-D RTS configuration and its related examples. Updated Appendix C – multicast channel configuration of derivatives market datafeed
V1.32	08 February 2014	Update OMD-D canned data to OMD-D Readiness Test Session 1 Data
V1.33	04 August 2014	Updated OMD-D canned data to the derivatives market production data
V1.34	8 October 2014	Included OMD-D canned data to the derivatives market for Asia Commodities Futures Contracts
V1.4	8 March 2016	Included OMD-C canned data to the securities market for Volatility Control Mechanism (VCM) and Closing Auction Session (CAS)
V2.0	10 December 2015	Updated to use PCAP file for replay
V2.1	31 March 2017	Include OMD-C SCM channels
V2.2	04 May 2017	Appendix B – Update info for OMD-C Index and OMD-C SCM channels
V2.3	15 Nov 2017	Appendix C – Update info for OMDD-Lite channels
V2.3.1	28 Nov 2017	Appendix C – Update info for OMDD-Lite Order Feed channels
V2.4	19 Mar 2018	Appendix B – Update info for OMD-C channels
V2.5	31 May 2018	Updated OMD-C packet rate for performance testing
V2.6	16 July 2020	New OMD-C processes resilience model

CONTENTS

1   Introduction

This document specifies the On-boarding Tools of the HKEX Orion Market Data Platform ("OMD") Securities Market & Index Datafeed Products. The On-boarding Tool of the HKEX Orion Market Data Platform ("OMD") Derivatives Market Datafeed Products is specifies in another document.

The OMD On-boarding tools provide testing facilities and simulated market data (canned data) for OMD Information Vendors and Exchange Participants (clients). This facilitates the development, enhancement and testing of the client's systems which receive and process data from OMD. The OMD On-boarding tools are intended to help clients ensure their readiness for the implementation before joining the Open Test, Readiness Test and Market Rehearsal (MR) as required by HKEX.
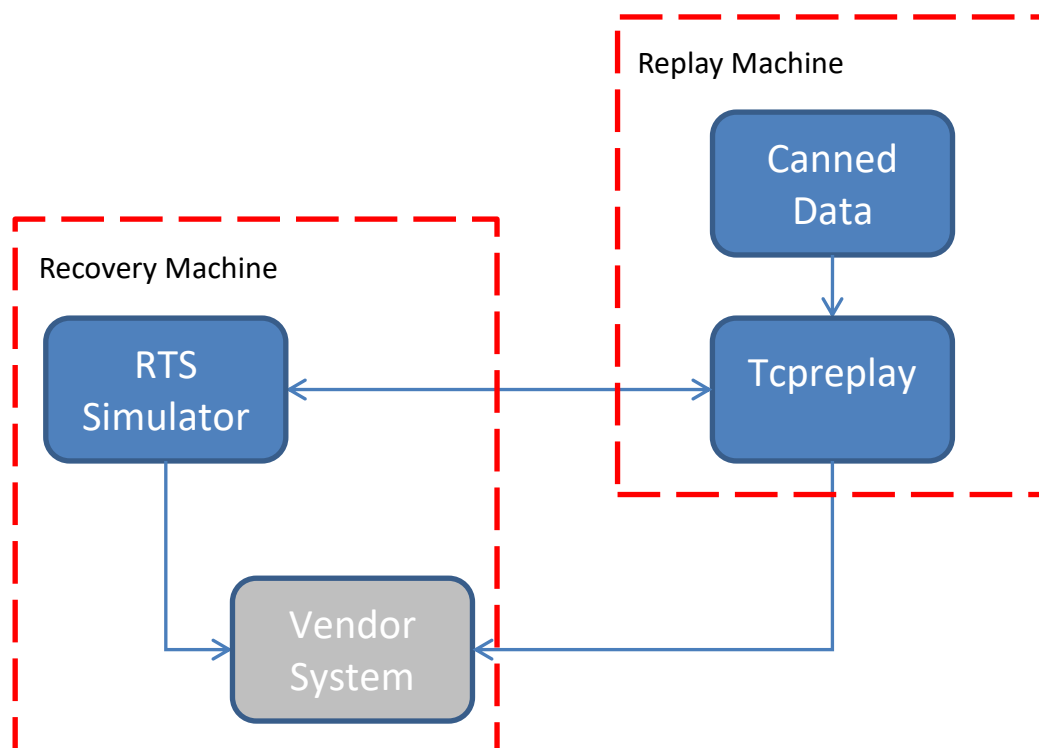
OMD On-boarding tools support clients' capacity and performance tests via allowing canned data to be replayed at various fixed rates. Clients can also make use of the OMD On-boarding tools for testing their own system development and enhancements. By distributing different set of canned data to the clients, HKEX aims to support the clients in testing their systems' readiness for various implementations of OMD initiatives.

Table 1. Acronyms used in this document

HKEX	Hong Kong Exchanges and Clearing Limited
OMD	Orion Market Data Platform
RTS	ReTransmission Servers
XDP	eXchange Data Publisher

1.1   System Relationship Diagram

The relationship diagram between the OMD and clients' application systems and the system's components are listed below.

1.2 OMD On-Boarding Tools Components

Component	Description
RTS Simulator	Facilitates testing of client's software on OMD retransmission services including retransmission request, retransmission response, logon, logon response and the processing of the missed messages requested as sent via unicast connection. There are 4 RTS simulators running by default to provide the resilience service.
tcpreplay	Broadcasts OMD canned data via multicast over the network The following functions are available: - Rate control in packets per second - Broadcasts on both A&B lines
Canned Data	Fictitious trading data to facilitate testing - Separate files may exist for different testing purpose (e.g. volume test or functional test) - The canned data will be injected by the tcpreplay to simulate the OMD production/testing environment.

2   System Requirement and Installation

2.1   Hardware Requirements

2 connected machines (Replay Machine and Recovery Machine) each with the following specification
- 64-bit AMD or Intel processor
- 16 GB RAM
- 20 GB available disk space

2.2   Software Requirements

- Red Hat Enterprise Linux (RHEL) 6.5 64-bit Edition, or CentOS 6.5 64-bit
- root / sudo access
- Tcpdump
- Tcpreplay (Replay Machine only)

2.3   Installation Procedures

(a)   <u>Replay Machine</u>

Step	Description
1.	Copy the 'cannedDataLocalizingTool.tar.gz' directory from the provided media to the desired host. Here is an example command: `cp -r /`**`*Path to CDROM*`**`/ cannedDataLocalizingTool.tar.gz /home/userid`
2.	Extract the TAR/GZIP file to the correct directory: `tar xvzf cannedDataLocalizingTool.tar.gz`

3.	To verify the installation was successful, the following command will show the contents of the install directory:
	
	`ls /home/userid/cannedDataLocalizingTool`
	
	If successful, the directory will contain:
	
	`common.sh pcapPreparationTool.sh preparePlaybackCacheFile.sh rewriteChecksum.sh updateIpMacAddress.sh`

4.	Install tcpreplay 3.4.4: Please refer to the installation instruction provided by tcpreplay.

(b)   Recovery Machine

Step	Description
1.	Copy the 'onboardingTools_2.0.tar.gz' directory from the provided media to the desired host. Here is an example command: `cp -r /*Path to CDROM*/onboardingTools_2.0.tar.gz /home/userid`
2.	Extract the TAR/GZIP file to the correct directory: `tar xvzf onboardingTools_2.0.tar.gz -C /opt` *Note* - the tools will not work if they are not in the correct directory structure.
3.	Add the correct directory permissions: `chown -R {userid:usergroup} /opt/hkex` Where userid & usergroup are system account that using to run the onboarding tools.

4.	To verify the installation was successful, the following command will show the contents of the install directory:
	
	`ls /opt/hkex/xdp/release/`
	
	If successful, the directory will contain:
	
	`bin config data docs lib logs profile.wombat`

2.4   Configuration

The On-boarding Tools package is designed to playback data from the Replay machine, and to run recovery service (RTS Simulator) on the Recovery machine. The client application can be run on the Recovery machine connected to recovery service via the loopback interface (a virtual software device created by the kernel) or can be run on separate machine connected to recovery service via TCP connection.

Since the network connection between Replay machine and Recovery machine is different in each setup, it is required to modify the recovery service configuration so as to fit the interface available on the Replay machine. This section will detail how to do that.

(a)   Configuration

The configuration for the RTS Simulators are stored in the files 'xdp-rts-client-sim-01.xml', 'xdp-rts-client-sim-02.xml', 'xdp-rts-client-sim-03.xml' and 'xdp-rts-client-sim-04.xml' under the 'config' directory. Each of the configuration file controls the corresponding RTS simulator and all the 4 simulators are running in parallel to provide the resilience service.
The configuration of the RTS simulators are similar and the difference is the open port (RtsTcpPort) to the client application.

Example 1, Default configuration of 'xdp-rts-client-sim-01.xml'.
The open port are **6969**, **6970**, **6971** and **6972** for different RTS simulator instances.

```xml
<!-- Multiple Machine Config Change -->

<!-- Point C -->

    <Parameter>
      <Name>RtsTcpIpAddress</Name>
      <Value>127.0.0.1</Value>
    </Parameter>

    <Parameter>
      <Name>RtsTcpPort</Name>
      <Value>6969</Value>
    </Parameter>

<!-- Point B -->

    <Parameter>
      <Name>DefaultMulticastInterface1</Name>
      <Value>127.0.0.1</Value>
    </Parameter>

    <Parameter>
      <Name>DefaultMulticastInterface2</Name>
      <Value>127.0.0.1</Value>
    </Parameter>

<!-- Point D -->

    <ClientList>
      <Client>
          <UserName>test01</UserName>
          <IpAddresses>127.0.0.1</IpAddresses>
          <MaximumGap>100</MaximumGap>
          <MaximumRequests>1000</MaximumRequests>
          <MaximumMessages>10000</MaximumMessages>
          <Parameter>
              <Name>Channels</Name>
              <Value>1</Value>
              <Value>2</Value>
              <Value>3</Value>
              <Value>4</Value>
              <Value>10</Value>
              <Value>20</Value>
              <Value>XX</Value>
              <Value>78</Value>
          </Parameter>
      </Client>
    </ClientList>

<!-- /Multiple Machine Config Change -->
```

Example 2, Configuration for client application on the Recovery Machine

```
<!-- Multiple Machine Config Change -->

<!-- Point C -->

    <Parameter>
      <Name>RtsTcpIpAddress</Name>
      <Value>127.0.0.1</Value>
    </Parameter>

    <Parameter>
      <Name>RtsTcpPort</Name>
      <Value>6969</Value>
    </Parameter>

<!-- Point B -->

   <Parameter>
      <Name>DefaultMulticastInterface1</Name>
      <Value>10.0.2.15</Value>
   </Parameter>

   <Parameter>
      <Name>DefaultMulticastInterface2</Name>
      <Value>10.0.2.15</Value>
   </Parameter>

<!-- Point D -->

    <ClientList>
       <Client>
           <UserName>test01</UserName>
           <IpAddresses>127.0.0.1</IpAddresses>
           <MaximumGap>100</MaximumGap>
           <MaximumRequests>1000</MaximumRequests>
           <MaximumMessages>10000</MaximumMessages>
           <Parameter>
               <Name>Channels</Name>
               <Value>1</Value>
               <Value>2</Value>
               <Value>3</Value>
               <Value>4</Value>
               <Value>10</Value>
               <Value>20</Value>
               <Value>XX</Value>
               <Value>78</Value>
           </Parameter>
       </Client>
   </ClientList>


<!-- /Multiple Machine Config Change -->
```

Example 3, Configuration for Client application not on Recovery Machine

```
<!-- Multiple Machine Config Change -->

<!-- Point C -->

    <Parameter>
      <Name>RtsTcpIpAddress</Name>
      <Value>10.0.3.15</Value>
    </Parameter>

    <Parameter>
      <Name>RtsTcpPort</Name>
      <Value>6969</Value>
    </Parameter>

<!-- Point B -->

    <Parameter>
      <Name>DefaultMulticastInterface1</Name>
      <Value>10.0.2.15</Value>
    </Parameter>

    <Parameter>
      <Name>DefaultMulticastInterface2</Name>
      <Value>10.0.2.15</Value>
    </Parameter>

<!-- Point D -->

    <ClientList>
       <Client>
           <UserName>test01</UserName>
           <IpAddresses>10.0.3.16</IpAddresses>
           <MaximumGap>100</MaximumGap>
           <MaximumRequests>1000</MaximumRequests>
           <MaximumMessages>10000</MaximumMessages>
           <Parameter>
               <Name>Channels</Name>
               <Value>1</Value>
               <Value>2</Value>
               <Value>3</Value>
               <Value>4</Value>
               <Value>10</Value>
               <Value>20</Value>
               <Value>XX</Value>
               <Value>78</Value>
           </Parameter>
       </Client>
    </ClientList>


<!-- /Multiple Machine Config Change -->
```
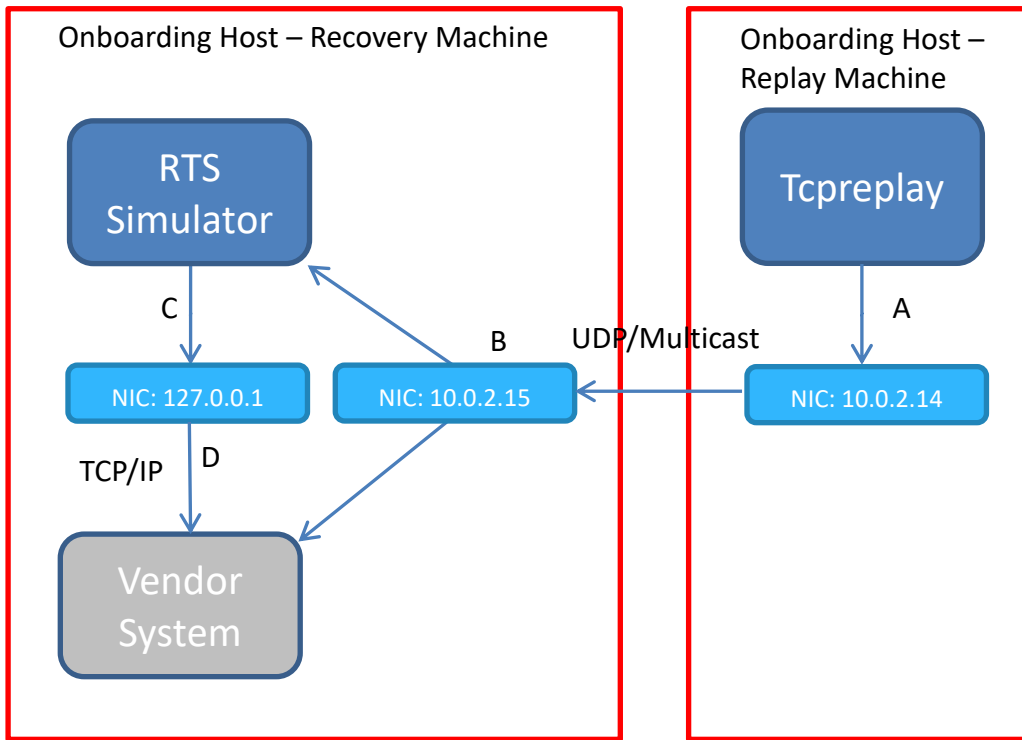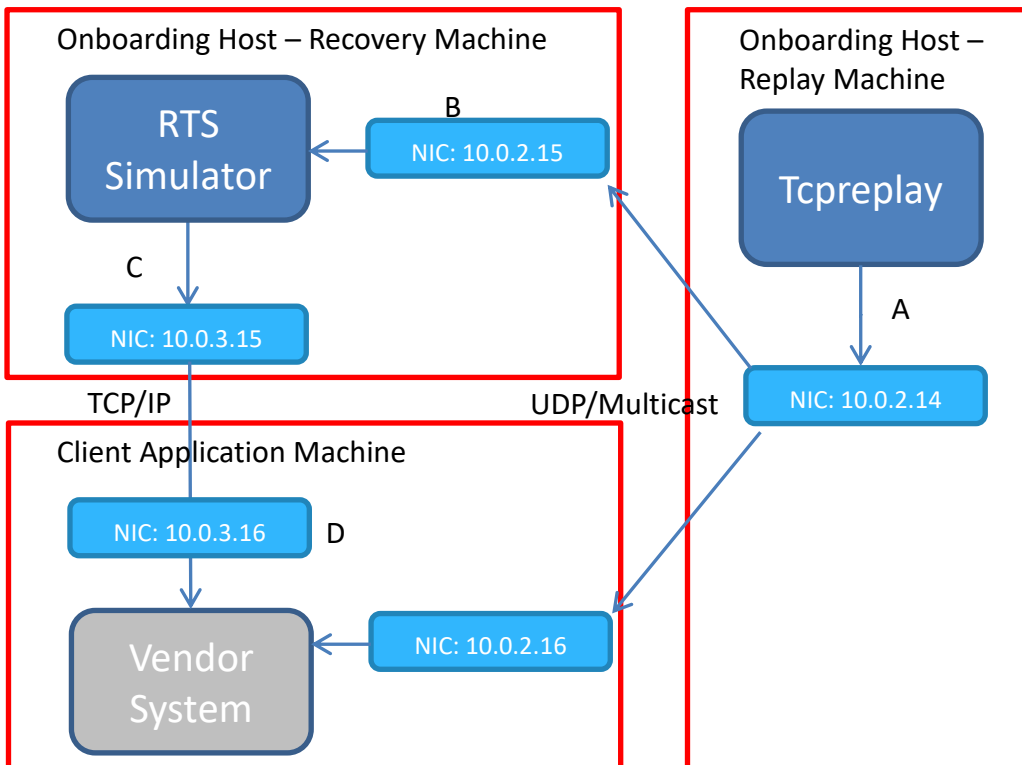
*XX - Denotes a range of channels defined in the configuration file, see the 'xdp-rts-client-sim-01.xml' for full details.

(b)   Diagram 1 – Client Application on Recovery Machine

(c)   Diagram 2 – Client Application not on Recovery machine

Step	Description
1.	Edit the RTS Simulator's configuration. Open the RTS configuration xml, at the top there is a multi-host section as detailed in section 2.4 a. Below is a description of the relevant parameters. **Point B** - This defines the network interface where the RTS Simulator listens for market data from the tcpreplay. This should nearly always be set to the interface connected to Point A, which is detailed in the next step. *DefaultMulticastInterface1* - The primary (A line) interface that the RTS Simulator listens to for multicast data from the tcpreplay. *DefaultMulticastInterface2* - The secondary (B line) interface that the RTS Simulator listens to for multicast data from the tcpreplay. This can be the same as the primary interface for this test. **Point C -** This is where the RTS Simulator interface to the client is defined, the IP address here can be any valid network interface on the host. *RtsTcpIpAddress* - The TCP/IP address that the RTS Simulator will listen for client connections on. *RtsTcpPort* - The TCP/IP port the RTS Simulator will listen for client connections on. Firstly, points B & C need to be defined. • For customers running client application on Recovery Machine please refer to Diagram 1 and Example 2. • For customers running client application on a separate machine please refer to Diagram 2 and Example 3. Once the values have been set accordingly save the configuration file. **Point D -** This is where the RTS Simulator list of clients is defined, the IP address here is per client and has to be the IP address that the corresponding user is going to login from. *IpAddress* - The RTS clients IP address.
2.	**Point A** - is the network interface that is the source of market data for both the client and the RTS Simulator.
3.	Once the script has been started subscribe to one of the channels with a test application to verify the changes are working.

3 Operational Procedures

3.1 Starting OMD On-boarding tools

Start OMD Environment on Recovery Machine

Step	Description
1.	Start in the OMD path: `cd /opt/hkex/xdp/release`

2.	Source the profile, this will set various environment variables which are based on the install path:
	
	`source profile.wombat`
	
	Change directory to the 'bin' directory:
	
	`cd bin`

3.	Run the start script to start ReTransmission Server: `start_simulation.sh` The script will output the following to the screen if successful: *Starting ReTransmission Server* *Logging to /opt/hkex/xdp/release//logs*
4.	To verify that the processes are running, check that their log files are updating via: `ls /opt/hkex/xdp/release/logs` Once these processes are running, it is possible to receive OMD data from the Replay Machine and utilize the retransmission server, which will be populated by the data being sent by the tcpreplay. At the end of the replay file, the tcpreplay will cease broadcasting data and the playback.sh script should be executed once more.

Play back Canned Data on Replay Machine

Step	Description
1.	Start in the Canned Data Localizing Tool path: `cd /home/userid/cannedDataLocalizingTool`

2.	Execute the Canned Data Localizing Script:
	
	`./pcapPreparationTool.sh <Canned Data File Name>`
	
	During the execution the script will ask runner to select the interface to be used for the data playback. After selection, a script "playback.sh" will be generated.
	
	The script will output the following to the screen if successful:
	
	`=================================================================`
	`NIC #     NIC       IP                    MAC Address`
	`=================================================================`
	`1        eth0      10.1.65.50            e8:39:35:eb:97:98`
	`2        eth1      10.1.88.50            e8:39:35:eb:97:9a`
	`3        eth2      10.0.8.50             e8:39:35:eb:97:9c`
	`4        eth3      10.1.89.50            e8:39:35:eb:97:9e`
	`5        eth4      10.1.66.50            f4:ce:46:a9:eb:8c`
	`6        eth5      10.70.12.50           f4:ce:46:a9:eb:8d`
	`7        eth6      10.1.90.50            f4:ce:46:a9:eb:8e`
	`8        eth7      10.0.9.50             f4:ce:46:a9:eb:8f`
	`=================================================================`
	`Input Source NIC # above for playback Line A and Line B data : 1`
	
	`Selected NIC (Line A) # eth0 10.1.65.50 e8:39:35:eb:97:98 (1)`
	`Selected NIC (Line B) # eth0 10.1.65.50 e8:39:35:eb:97:98 (1)`
	
	`############################################################################`
	`################################################`
	`# Start preparing the pcap files`
	`############################################################################`
	`################################################`
	
	`1. Update the source IP and MAC address`
	`tcprewrite --pnat=10.1.65.0/24:10.1.65.50 --enet-smac=e8:39:35:eb:97:98 --`
	`infile=pcap/omdc-20130913.pcap --outfile=pcap/omdc-20130913.pcap.tmp`
	`tcprewrite --pnat=10.1.66.0/24:10.1.66.50 --enet-smac=f4:ce:46:a9:eb:8c --`
	`infile=pcap/omdc-20130913.pcap.tmp --outfile=pcap/omdc-20130913.pcap PLAYBACK`
	
	`2. Recalculate IPv4/TCP/UDP header checksums`
	`tcprewrite -C -i pcap/omdc-20130913.pcap_PLAYBACK -o pcap/omdc-`
	`20130913.pcap_PLAYBACK.rewrite`
	
	`3. Prepare cache file for tcpreplay to playback Line A / B`
	`tcpprep -e e8:39:35:eb:97:98 -i pcap/omdc-20130913.pcap PLAYBACK.rewrite -o`
	`pcap/omdc-20130913.pcap_PLAYBACK.rewrite.tag`
	
	`############################################################################`
	`################################################`
	`Update pcap file completed SUCCESFULLY.`
	
	`You can now playback the file with ./playback.sh`
	
	`=> ./playback.sh (sudo tcpreplay -i eth0 -I eth4 -x 10 -c pcap/omdc-`
	`20130913.pcap PLAYBACK.rewrite.tag pcap/omdc-20130913.pcap PLAYBACK.rewrite)`
	
	Note - the Canned Data File can be found in the root path of the provided media, with extension "pcap".

3.	Playback data using the generated playback script. `./playback.sh`

3.2   Retransmission Server

The retransmission server simulators are started via the 'start_simulation.sh' script. The server is communicated with via a simple TCP/IP protocol.

The retransmission servers operates using IP: 127.0.0.1, Port: 6969 / 6970/ 6971 / 6972 and UserName: test01.

3.3   Log Files

Log files only serve to indicate that the process is up (it is still writing to the log file) and how much data it is receiving.

Log files can be found in the 'logs' directory.

Each Retransmission Server produces two log files, takes server instance 1 as example, the log files are 'xdp-rts-01.log' and 'xdp-rts-01.log1'. Most recent information is kept in the 'xdp-rts-01.log' and then rolled into the 'xdp-rts-01.log1' file when it is updated.

Example 4, Retransmission Server Log: xdp-rts-01.log

```
*****************************************************************
17:39:29 Info: xdp-rts:  Version ?.?.?  Date date  Build   Components
17:39:29 Info: *****************************************************************
17:39:29 Info: Started at:         Fri Jul 20 17:39:29 2012
17:39:29 Info: Hostname:           localhost
17:39:29 Info: Configuration file: xdp-rts-client-sim-01.xml
```

Example 5, Retransmission Server Log: xdp-rts-01.log

```
16:24:50 Info:      ------------------------+----------+----------+--------------+
16:24:50 Info:  2012/10/17 16:24:50   Wimp |    Rate  |     Max  |       Total  |
16:24:50 Info:      ------------------------+----------+----------+--------------+
16:24:50 Info:    LineDataBlock.xdp-rts1  |       0  |    1620  |        1620  |
16:24:50 Info:    LineDataBlock.xdp-rts2  |       0  |     102  |         102  |
16:24:50 Info:    LineDataBlock.xdp-rts3  |       0  |    2000  |       13076  |
16:24:50 Info:    LineDataBlock.xdp-rts4  |       0  |      20  |          20  |
16:24:50 Info:   LineDataBlock.xdp-rts10  |       0  |    2001  |      102566  |
16:24:50 Info:   LineDataBlock.xdp-rts20  |       0  |    2000  |       23548  |
16:24:50 Info:   LineDataBlock.xdp-rts21  |       0  |    2000  |       22910  |
16:24:50 Info:   LineDataBlock.xdp-rts22  |       0  |    2001  |      104802  |
16:24:50 Info:   LineDataBlock.xdp-rts23  |       0  |    2000  |       14292  |
16:24:50 Info:   LineDataBlock.xdp-rts24  |       0  |    2001  |       31912  |
16:24:50 Info:   LineDataBlock.xdp-rts25  |       0  |      20  |          20  |
16:24:50 Info:   LineDataBlock.xdp-rts26  |       0  |      20  |          20  |
16:24:50 Info:   LineDataBlock.xdp-rts27  |       0  |      20  |          20  |
16:24:50 Info:   LineDataBlock.xdp-rts28  |       0  |      20  |          20  |
16:24:50 Info:   LineDataBlock.xdp-rts30  |       0  |    2001  |       22382  |
16:24:50 Info:   LineDataBlock.xdp-rts31  |       0  |    2000  |       21806  |
16:24:50 Info:   LineDataBlock.xdp-rts32  |       0  |    2001  |      100328  |
16:24:50 Info:   LineDataBlock.xdp-rts33  |       0  |    2000  |       14460  |
16:24:50 Info:   LineDataBlock.xdp-rts34  |       0  |    2001  |       31982  |
16:24:50 Info:   LineDataBlock.xdp-rts35  |       0  |      20  |          20  |
16:24:50 Info:   LineDataBlock.xdp-rts36  |       0  |      20  |          20  |
16:24:50 Info:   LineDataBlock.xdp-rts37  |       0  |      20  |          20  |
16:24:50 Info:   LineDataBlock.xdp-rts38  |       0  |      20  |          20  |
16:24:50 Info:   LineDataBlock.xdp-rts41  |       0  |    2001  |      215642  |
16:24:50 Info:   LineDataBlock.xdp-rts42  |       0  |    2000  |       29930  |
16:24:50 Info:   LineDataBlock.xdp-rts43  |       0  |    2000  |        6656  |
16:24:50 Info:   LineDataBlock.xdp-rts60  |       0  |    2001  |       64918  |
16:24:50 Info:   LineDataBlock.xdp-rts70  |       0  |    1160  |        1160  |
16:24:50 Info:   LineDataBlock.xdp-rts71  |       0  |    1170  |        1170  |
16:24:50 Info:   LineDataBlock.xdp-rts72  |       0  |    2000  |        4942  |
16:24:50 Info:   LineDataBlock.xdp-rts73  |       0  |     702  |         702  |
16:24:50 Info:   LineDataBlock.xdp-rts74  |       0  |    1704  |        1704  |
16:24:50 Info:   LineDataBlock.xdp-rts75  |       0  |      20  |          20  |
16:24:50 Info:   LineDataBlock.xdp-rts76  |       0  |      20  |          20  |
16:24:50 Info:   LineDataBlock.xdp-rts77  |       0  |      20  |          20  |
16:24:50 Info:   LineDataBlock.xdp-rts78  |       0  |      20  |          20  |
16:24:50 Info:      ------------------------+----------+----------+--------------+
```

We have highlighted three key log messages here to search for in the files as the initialization is particularly long.

Example 4 indicates the successful starting of the process, which host it is running on, and that it has found its configuration file.

Example 5 shows that the Retransmission server is receiving data on all of the 36 channels.

Remark: The Retransmission Server Log primarily facilitates HKEX checking when there are queries related to the OMD retransmission service in the On-boarding tools raised by the Clients. The two examples quoted here are sufficient for the Clients to health check the normal running of the Retransmission Server, Clients are not required to check other messages as logged in the Log file.

3.4   Changing Market Data

It may be desirable to switch between various sets of market data that we have released.

This can be achieved by re-running the Canned Data Localization Script with the new data as described in section 3.1, and then replaying data using the newly generated playback.sh.

3.5   Shutdown of OMD

The shutdown of OMD is very simple. A script has been provided to shut down all tools.

Step	Description
1.	Run the following: `stop_simulation.sh`

3.6   Performance Testing

Tcpreplay can control the message rate at which the canned data file is replayed. Therefore, the OMD On-boarding tools are also capable of being used to test system performance, namely throughput. Clients can modify the playback.sh generated by the Canned Data Localizing script to add argument for specifying the packet rate (--pps), replay multiplier (--multiplier) or the playback bandwidth (--mbps) in the play back.

Clients can use the following table as a guide to set the packet rate in their testing:

Datafeed	Packet Rate	Expected Message Volume (bandwidth)
SS	905	16Mbps
SP	2497	30Mbps
SF	7362	60Mbps
Index	164	1Mbps

It is expected that before the Readiness Test that clients have stressed their systems to the equivalent message bandwidth of the market data package that they have chosen.

The packet rates recommended for different OMD Datafeed Products for testing the client system in handling the stated message volume are for reference only. It may not drive the expected message volume but will be as close as possible. The recommended packet rates only apply to the canned data we provided and may be different for different sets of canned data.

Remarks:

1. Multiple logons to the RTS Simulator using the same socket is not supported by the On-boarding Tools.
2. Clients should avoid using two applications (or two instances of the same application) to log-in the RTS Simulator using the same UserName which will result in all further login attempts on that UserName being refused.
3. Please refer to tcpreplay documentation regarding the control of packet replay rate.

Appendix A – FAQ and Troubleshooting

Reporting Issues

In the event that the software is not working as expected please verify the following:

- Run 'uname -a' to ensure the Red Hat or Cent OS version is 6.2.
- Run 'free -m' and ensure there is at least 15GB free memory, and 16GB total system RAM
- Run the package on its default configuration and verify data flow as described in Section 3.

Include the results of these tests in your email to the support group.

FAQ

1. **Why our xdp-rts process dies immediately after startup?**
   This is usually caused by not enough free memory; occasionally the OS will report this in the 'dmesg' log, but not always. The xdp-rts process allocates a large amount of its memory on initialization.

2. **We encountered 'undefined symbol', 'symbol lookup error' and 'library not found', how can we proceed?**
   These errors are found when the package is installed on the wrong version of Red Hat or Cent OS.

3. **Why our login is rejected?**
   This is possibly due to the following reasons:
   - *the packet header is not sent as required at the start of the login string or*
   - *the packet size is sent big endian instead of little endian per Interface Specification or*
   - *the packet is space terminated instead of NULL terminated.*

   Example valid login hex string:

   20 00 01 30 01 00 00 00 00 00 00 00 00 00 00 00 10 00 65 00 74 65 73 74 30 31 00 00 00 00 00 00

   The packet header is highlighted.

   Questions related to Network setup

   i. **Client Application on Recovery Machine**

4. **Why our system can receive data but RTS Simulator responding with 'MESSAGES_NOT_AVAILABLE'?**
   The RTS Simulator is still configured to listen to loopback address

   Please verify that both the "Point A" and "Point B" are set to the same value in the RTS configuration xml files.

5. **Why our system is unable to connect to the RTS Simulator or login rejected?**
   The "Point C" or "Point D" IP address has been updated incorrectly in the RTS configuration xml files.

   Please verify that the "Point C" IP address has been updated correctly in the RTS configuration xml files and is not currently set to the loopback address. Also please verify that the "Point D" IP address has been updated correctly in the RTS configuration xml files and is not currently set to the loopback address. This address should be set to the source address of the connection to the retransmission server.

6. **Why the RTS Simulator rejects the login from our system?**
   Please refer to Question 5 above for details.

ii.   <u>**Client Application not on Recovery Machine**</u>

7. **Why our system can receive data but RTS Simulator responding with 'MESSAGES_NOT_AVAILABLE'?**
   This is possibly due to the following reasons:
   - *The RTS Simulator still configured to listen to loopback address, verify that both the "Point A" and "Point B" are set to the same value in the RTS configuration xml files. In this setup the value for "Point C" is likely different from A & B. The RTS Simulator still needs to listen to traffic from the tcpreplay, even if responses are sent and received sent via another so A and B need to be the same interface.*
   - *The RTS Simulator still configured to listen an address that cannot consume the tcpreplay traffic, verify that both the "Point A" and "Point B" are set to the same value in the RTS configuration xml files.*

8. **Why our system is unable to connect to the RTS Simulator or login rejected?**
   The "Point C" or "Point D" IP address has been updated incorrectly in the RTS configuration xml files.

   Please verify that the Point C IP address has been updated correctly in the RTS configuration xml files and is not currently set to the loopback address. Also please verify that the Point D IP address has been updated correctly in the RTS configuration xml files and is not currently set to the loopback address. This address should be set to the source address of the connection to the retransmission request.

9. **Why the RTS Simulator rejects the login from our system?**
   Please refer to Question 8 above for details.

Appendix B – Multicast Channel Configuration (Securities Market, Index Datafeed, and Stock Connect Market)

The multicast channel IDs presented here are for testing purposes only. Clients should not assume that the multicast channel IDs in production will be identical to those presented here.

Channel ID	Multicast Address & Port	Message Type	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)	OMD Stock Connect Market (SCM)
1	239.1.1.1/51000 239.1.127.1/51000	Market Definition (10) Security Definition (11) Liquidity Provider (13) Currency Rate(14)	●	●	●		
2	239.1.1.1/51001 239.1.127.1/51001	Trading Session Status (20)	●	●	●		
3	239.1.1.1/51002 239.1.127.1/51002	Market Turnover (61)	●	●			
4	239.1.1.1/51003 239.1.127.1/51003	News (22)	●	●			
9	239.1.1.1:51008 239.1.127.1:51008	Disaster Recovery Signal (105)	●	●	●		

No	Multicast Address	Message Types					
10	239.1.1.2/51000 239.1.127.2/51000	Trade Ticker (52) Nominal Price (40) Indicative Equilibrium Price (41) Closing Price (62) Aggregate Order Book Update (53) Statistics (60) Yield (44) Broker Queue (54) VCMTrigger (23) Reference Price (43) Order Imbalance (56) Security Status (21)	●				
20	239.1.1.3/51000 239.1.127.3/51000	Trade (50) Trade Cancel (51) Nominal Price (40) Indicative Equilibrium Price (41) Closing Price (62) Aggregate Order		●			
21	239.1.1.3/51001 239.1.127.3/51001			●			
22	239.1.1.3/51002 239.1.127.3/51002			●			
23	239.1.1.3/51003 239.1.127.3/51003			●			
24	239.1.1.3/51004 239.1.127.3/51004			●			
25	239.1.1.3/51005 239.1.127.3/51005			●			
26	239.1.1.3/51006 239.1.127.3/51006			●			
27	239.1.1.3/51007 239.1.127.3/51007			●			

28	239.1.1.3/51008 239.1.127.3/51008	Book Update (53) Statistics (60) Yield (44) VCM Trigger (23) Reference Price (43) Order Imbalance (56) Security Status (21)		●			
30	239.1.1.4/51000 239.1.127.4/51000	Trade (50) Trade Cancel (51) Add Order (30) Modify Order (31) Delete Order (32) Indicative Equilibrium Price (41) VCM Trigger (23) Reference Price (43) Order Imbalance (56) Security Status (21)			●		
31	239.1.1.4/51001 239.1.127.4/51001				●		
32	239.1.1.4/51002 239.1.127.4/51002				●		
33	239.1.1.4/51003 239.1.127.4/51003				●		
34	239.1.1.4/51004 239.1.127.4/51004				●		
35	239.1.1.4/51005 239.1.127.4/51005				●		
36	239.1.1.4/51006 239.1.127.4/51006				●		
37	239.1.1.4/51007 239.1.127.4/51007				●		
38	239.1.1.4/51008 239.1.127.4/51008				●		
41	239.1.1.5/51000 239.1.127.5/51000	Index Definition (70) Index Data (71)				●	
42	239.1.1.6/51000 239.1.127.6/51000					●	
43	239.1.1.7/51000 239.1.127.7/51000					●	
44	239.1.1.7/51001 239.1.127.7/51001					●	
49	239.1.1.7:51008 239.1.127.7:51008	Disaster Recovery Signal (105)				●	

60	239.1.1.8/51000 239.1.127.8/51000	Broker Queue (54)		▲	▲		
70	239.1.1.9/51000 239.1.127.9/51000	Add Odd Lot Oder (33) Delete Odd Lot Order (34)	▲	▲	▲		
71	239.1.1.9/51001 239.1.127.9/51001		▲	▲	▲		
72	239.1.1.9/51002 239.1.127.9/51002		▲	▲	▲		
73	239.1.1.9/51003 239.1.127.9/51003		▲	▲	▲		
74	239.1.1.9/51004 239.1.127.9/51004		▲	▲	▲		
75	239.1.1.9/51005 239.1.127.9/51005		▲	▲	▲		
76	239.1.1.9/51006 239.1.127.9/51006		▲	▲	▲		
77	239.1.1.9/51007 239.1.127.9/51007		▲	▲	▲		
78	239.1.1.9/51008 239.1.127.9/51008		▲	▲	▲		
80	239.1.1.121/51000 239.1.127.121/51000	Stock Connect Daily Quota Balance (80) Stock Connect Market Turnover (81)					▲
81	239.1.1.121/51001 239.1.127.121/51001	Stock Connect Market Turnover (81)					▲

● Product-related channel
▲ Complimentary service

Examples of multicast channels to be subscribed by clients for different OMD products:
- SS clients need to subscribe to multicast channels #1, #2, #3, #4 & #10 for full set of SS market data.
- SP clients need to subscribe to multicast channels #1, #2, #3, #4 & #20 - 28 for full set of SP market data
- SF clients need to subscribe to multicast channels #1, #2 & #30 - 38 for full set of SF market data
- Index clients need to subscribe to multicast channels #41, #42, #43 & #44for full set of Index market data
- Clients subscribing SP, SF & Index products need to subscribe to multicast channels #1, #2, #3, #4, #20 - 28, #30 - 38, #41, #42, #43 & #44
- Client subscribing SS, SP, SF products need to subscribe multicast channel #9 to receive the DR signal
- Client subscribing Index product need to subscribe multicast channel #49 to receive the DR signal
-

Note: Though the canned data created follows the production configuration, depending on the order and trade activities, some channels (Channels 4, 23-28, 33-38) may have minimal or no trading data just heartbeats. This is expected behaviour.