NASTEL®

Driving Business Transaction Performance

AP/SR 100.001

AutoPilot® MQSonar for WebSphere MQ

Quick Start Guide

Member of AutoPilot® On-Demand for WebSphere MQ Family

Freemium Edition

Version 1.0

PUBLISHED BY:

RESEARCH & DEVELOPMENT

NASTEL TECHNOLOGIES, INC.

48 SOUTH SERVICE ROAD, SUITE 205

MELVILLE, NY 11747

DOCUMENT TITLE: **AUTOPILOT MQSONAR FOR WEBSPHERE MQ QUICK START GUIDE**

VERSION: **1.0**

DOCUMENT RELEASE DATE: **AUGUST 2012**

NASTEL DOCUMENT NUMBER: **AP/SR 100.001**

CONFIDENTIALITY STATEMENT: THE INFORMATION WITHIN THIS MEDIA IS PROPRIETARY IN NATURE AND IS THE SOLE PROPERTY OF NASTEL TECHNOLOGIES, INC. ALL PRODUCTS AND INFORMATION DEVELOPED BY NASTEL ARE INTENDED FOR LIMITED DISTRIBUTION TO AUTHORIZED NASTEL EMPLOYEES, LICENSED CLIENTS, AND AUTHORIZED USERS. THIS INFORMATION (INCLUDING SOFTWARE, ELECTRONIC AND PRINTED MEDIA) IS NOT TO BE COPIED OR DISTRIBUTED IN ANY FORM WITHOUT THE EXPRESSED WRITTEN PERMISSION FROM NASTEL TECHNOLOGIES, INC.

ACKNOWLEDGEMENTS:

THE FOLLOWING TERMS ARE TRADEMARKS OF NASTEL TECHNOLOGIES CORPORATION IN THE UNITED STATES OR OTHER COUNTRIES OR BOTH: TRANSACTIONWORKS, AUTOPILOT M6, AUTOPILOT/IT, AUTOPILOT M6 FOR WMQ, AUTOPILOT/WMQ, M6 WEB SERVER, AUTOPILOT/WEB, M6 WEB CONSOLE, MQCONTROL, MQCONTROL EXPRESS, AUTOPILOT/TRANSACTION MONITOR, AUTOPILOT/WAS, AUTOPILOT/OS MONITOR

THE FOLLOWING TERMS ARE TRADEMARKS OF THE IBM CORPORATION IN THE UNITED STATES OR OTHER COUNTRIES OR BOTH: IBM, MQ, MQSERIES, WEBSPHERE, WEBSPHERE MQ WIN-OS/2, AS/400, OS/2, DB2, AND AIX, Z/OS

THE FOLLOWING TERMS ARE TRADEMARKS OF HEWLETT-PACKARD IN THE UNITED STATES OR OTHER COUNTRIES OR BOTH: OPENVIEW, HP-UX

COMPAQ, THE COMPAQ LOGO, ALPHASERVER, COMPAQ INSIGHT MANAGER, CDA, DEC, DECNET, TRUCLUSTER, ULTRIX, AND VAX REGISTERED IN U.S. PATENT AND TRADEMARK OFFICE. ALPHA AND TRU64 ARE TRADEMARKS OF COMPAQ INFORMATION TECHNOLOGIES GROUP, L.P IN THE UNITED STATES AND OTHER COUNTRIES

SNMPC, SNMPC, WORKGROUP, AND SNMPC ENTERPRISE ARE TRADEMARKS OF CASTLE ROCK COMPUTING IN THE UNITED STATES OR OTHER COUNTRIES, OR BOTH.

SUN, SUN MICROSYSTEMS, THE SUN LOGO, IFORCE, JAVA, NETRA, N1, SOLARIS, SUN FIRE, SUN RAY, SUNSPECTRUM, SUN STOREDGE, SUNTONE, THE NETWORK IS THE COMPUTER, ALL TRADEMARKS AND LOGOS THAT CONTAIN SUN, SOLARIS, OR JAVA, AND CERTAIN OTHER TRADEMARKS AND LOGOS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF SUN MICROSYSTEMS, INC. IN THE UNITED STATES AND OTHER COUNTRIES.

INSTALLANYWHERE IS A REGISTERED TRADEMARK OF ZEROG SOFTWARE IN THE UNITED STATES OR OTHER COUNTRIES, OR BOTH.

THIS PRODUCT INCLUDES SOFTWARE DEVELOPED BY THE APACHE SOFTWARE FOUNDATION (HTTP://WWW.APACHE.ORG/). THE JAKARTA PROJECT" AND "TOMCAT" AND THE ASSOCIATED LOGOS ARE REGISTERED TRADEMARKS OF THE APACHE SOFTWARE FOUNDATION

INTEL, PENTIUM AND INTEL486 ARE TRADEMARKS OR REGISTERED TRADEMARKS OF INTEL CORPORATION IN THE UNITED STATES, OR OTHER COUNTRIES, OR BOTH

MICROSOFT, WINDOWS, WINDOWS NT, WINDOWS XP, AND THE WINDOWS LOGOS ARE REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION.

UNIX IS A REGISTERED TRADEMARK IN THE UNITED STATES AND OTHER COUNTRIES LICENSED EXCLUSIVELY THROUGH X/OPEN COMPANY LIMITED.

"LINUX" AND THE LINUX LOGOS ARE REGISTERED TRADEMARKS OF LINUS TORVALDS, THE ORIGINAL AUTHOR OF THE LINUX KERNEL. ALL OTHER TITLES, APPLICATIONS, PRODUCTS, AND SO FORTH ARE COPYRIGHTED AND/OR TRADEMARKED BY THEIR RESPECTIVE AUTHORS.

SCO CUSA, SCO DOCTOR, SCO DOCTOR FOR NETWORKS, SCO DOCTOR LITE, SCO GLOBAL ACCESS, SCO MPX, SCO MULTIVIEW, SCO NIHONGO OPENSERVER, SCO OK, THE SCO OK LOGO, SCO OPENSERVER, SCO OPEN SERVER, SCO PORTFOLIO, SCO POS SYSTEM, SCO TOOLWARE, AND THE WORLD NEVER STOPS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF CALDERA INTERNATIONAL, INC. IN THE U.S.A. AND OTHER COUNTRIES, ALL RIGHTS RESERVED.

ORACLE® IS A REGISTERED TRADEMARK OF ORACLE CORPORATION AND/OR ITS AFFILIATES

OTHER COMPANY, PRODUCT, AND SERVICE NAMES, MAY BE TRADEMARKS OR SERVICE MARKS OF OTHERS.

Contents

Figures

Tables

Chapter 1: Introduction

Welcome to the *Nastel AutoPilot® MQSonar WebSphere MQ Quick Start Guide*. This guide describes the installation, configuration and usability of MQSonar utility. Please review this guide carefully before using the product.

1.1 How This Guide is Organized

Chapter 1:	Identifies the users and history of the document and supplies support and reference information.
Chapter 2:	Contains a brief description of the MQSonar.
Chapter 3:	Provides instructions for downloading and deploying the MQSonar.
Chapter 4:	Explains how to run the MQSonar.
Chapter 5:	Explains how to optionally integrate the MQSonar with AutoPilot.
Chapter 6:	Provides troubleshooting information.
Index:	Contains an alphanumeric cross-reference of all topics and subjects of importance.

1.2 History of This Document

Table 1-1. Document History

Release Date	Document Number	Version	Summary
August 2012	AP/SR 100.000	1.0	Initial release

1.3 Release Notes

See **`README.htm`** files on your installation.

1.4 Intended Audience

This document is intended for personnel deploying, running, administering, and maintaining IBM's WebSphere MQ. The user should be familiar with:

- Target operating system environment
- Procedures for installing software on the target platform
- Configuring and using WebSphere MQ Queue Managers.

1.4.1 User Feedback

Nastel encourages all users to participate and submit comments and suggestions to the community forum.

1.5 Technical Support

Technical support for AutoPilot MQSonar for WebSphere MQ is provided by a community forum. To access the forum, use the link on the main page or you can link directly using http://www.nastel.com/support/mqsonar/.

This page intentionally left blank.

Chapter 2: About MQSonar

2.1 Introduction

The purpose of the MQSonar utility is to measure various times related to queue manager message flow. Much like SONAR, this is completed by sending a ping which is picked up by the target application and is echoed back. By measuring the time spent it can determine various characteristics of the message path. In WebSphere MQ terms, it sends messages to a listening application which replies with a response to those messages. MQSonar captures various times along the message path as well as the round-trip times.

The echo application can be the WebSphere MQ command server or the MQSonar echo program.

- When using the command server, MQSonar can be run without any changes to the queue manager. It does this by leveraging the ability of queue manager to receive and reply to administrative messages, often referred to as PCF. However, this requires that the utility to be run under an id with administrative privileges.
- When using the MQSonar echo program to generate the replies, minimal changes to the queue manager are required. In many cases, these changes may already be in place and no actual changes are required. The advantage of using the echo program is that it can be run by any user that has put/get privileges.

	NOTE:	PCF is IBM's Programmable Command Format. You do not need to know how to use PCF to use MQSonar.

MQSonar creates a batch of messages and submits them to WebSphere MQ. The queue used can be configured as local, alias, cluster or remote queue across one or more queue managers as long as it points eventually to the target queue being processed by the echo application.

Users can execute as many instances of MQSonar as needed, to collect statistics about different queue performance. The following are some basic configurations that can be implemented using MQSonar:

- Single queue manager (Figure 2-1):
  - determine if the queue manager is up and responding
  - determine standard throughput for a (set of) queues
  - compare the behavior of different queue configurations such as using persistent versus non-persistent messages

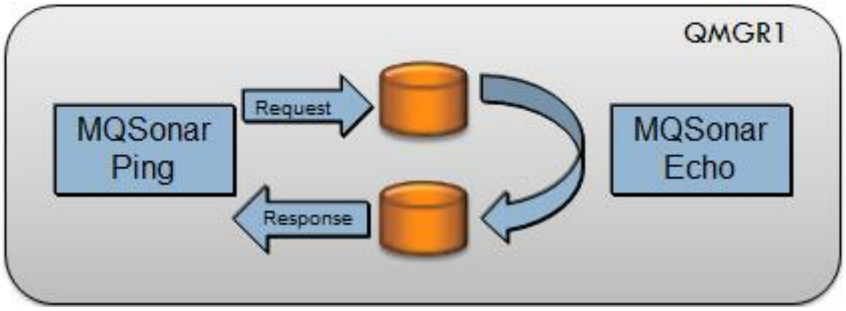


Figure 2-1. Single Queue Manager Flow

- Multiple queue managers (Figures 2-2 and 2-3):
    - Identify slowdowns in inter-queue manager communication
    - Identify queue managers that contribute to delays
    - Identify differences in behaviors of different queue managers
    - Verify that a path from one sending application to the receiving application is properly configured.

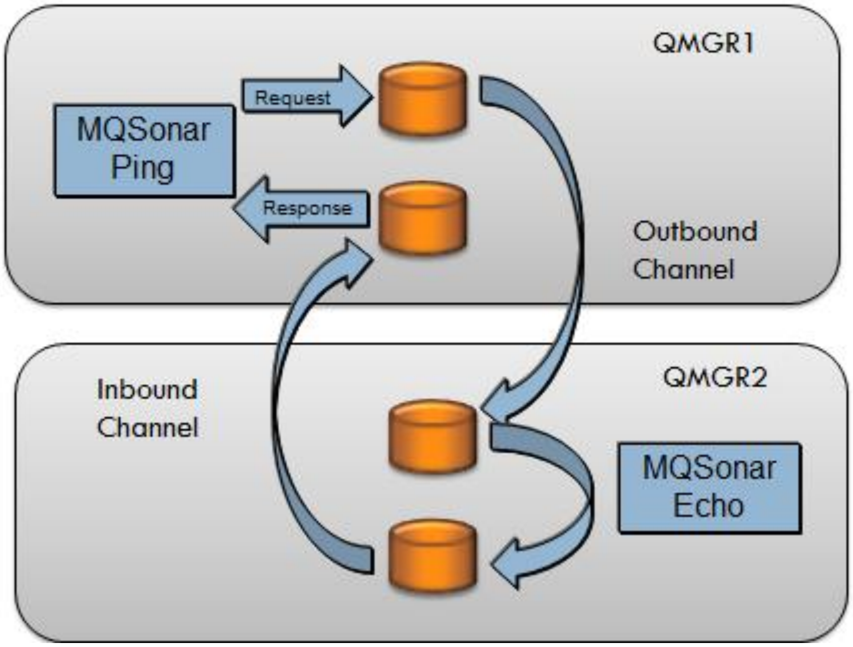


Figure 2-2. Two Queue Manager Flow Example

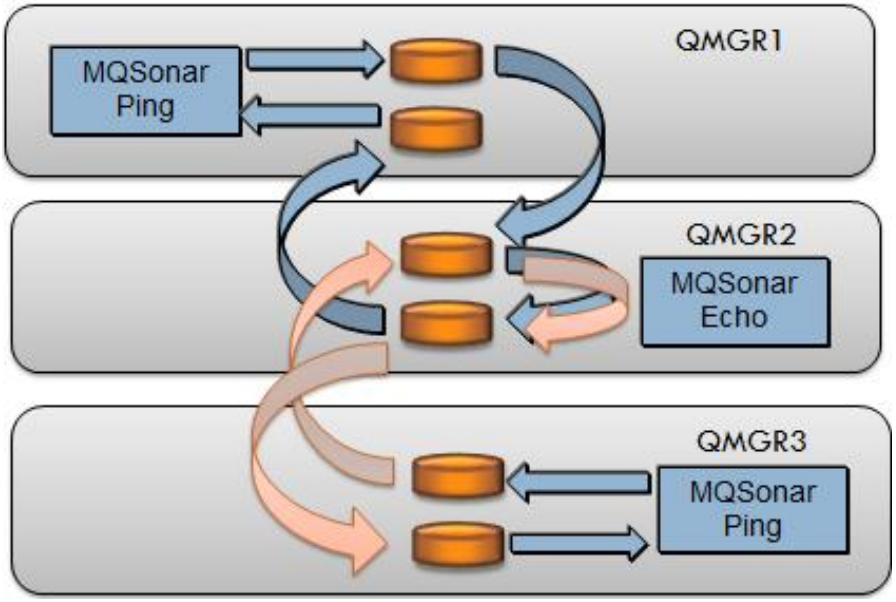


Figure 2-3. Multiple Queue Manager Flow Example

As a standalone utility, MQSonar displays it output to the console which can be captured and reviewed, and processed by scripts to provide a more permanent record of activity.

MQSonar can optionally be integrated with Nastel AutoPilot to provide a more comprehensive solution, including alerting when abnormal conditions occur, providing graphical views of the environment, and capturing historical trending. This is covered in Chapter 5, Integrating MQSonar with AutoPilot M6.

This page intentionally left blank.

Chapter 3: Installing MQSonar

3.1 Pre-requisites

MQSonar requires:

- Target queue manager running
- WebSphere MQ command server
  - Administrative permission to submit messages to command input queue
  - Authority to issue the PCF MQCMD_PING_Q_MGR command

or

- One or more instances of the MQSonar echo application executing
  - Authority to put and get messages to the request and response queues.

⚠	**IMPORTANT!**	When measuring message performance over remote queues: local and remote queue managers must have pair of sender/receiver channels defined and active that allows local and remote queue manager to exchange messages.

3.1 Installing MQSonar on Windows

The installation is provided as a simple zip file which can be installed on any location on your system. After installation, you execute one or both of the provided utilities (mqsonar,cmd, mqsonar_echo.cmd, nsqping and nsqecho).

3.2 Installing MQSonar on Linux

The installation is provided as a simple tar file which can be installed on any location on your system. After installation, you execute one or both of the provided utilities (mqsonar,sh, mqsonar_echo.sh, ./nsqping and ./nsqecho).

tar –xvf MQSonar_*version*_LINUX_RHEL_x86_64.tar

will create an mqsonar folder in the current folder.

3.3 Configuring Queue Names

By default, MQSonar uses a number of queues, including a request queue, a reply queue and a model to create the reply queue. The names of all of these queues can be changed if required, such as when running with limited authority. The default values are defined in the mqqual.ini file which is included in the config/groups folder in the installation materials:

- SYSTEM.DEFAULT.MODEL.QUEUE
  - Can be specified with the parameter NASTEL.MODEL.QUEUE= in mqqual.ini
- MQSONAR.ECHO.REPLY.nnn (temporary dynamic reply queue)
  - Can be specified with the HLQ and MLQ parameters in mqqual.ini

This page intentionally left blank.

Chapter 4: Running MQSonar

4.1 Executing the Echo Component

If you are going to use the WebSphere MQ Command server to provide the ping, then other than the authority and channel as discussed above, no other setup is required and you can continue with Executing the Ping Component. To use the MQSonar echo application, change to the folder where MQSonar is installed above and run the mqsonar_echo script (mqsonar_echo.cmd or mqsonar_echo.sh). When running the provided echo utility, the setup is described below (full details are in section 4.6, NSQRPL Command Line Options):

```
mqsonar_echo QueueName Qmgr_name
```

- **Example 1:** Listen on ECHO.QUEUE on queue manager QM2

```
mqsonar_echo ECHO.QUEUE QM2
```

- **Example 2:** Listen on TEST.QUEUE on queue manager QM1

```
./mqsonar_echo.sh TEST.QUEUE QM2
```

NOTE: The echo component cannot typically be a user application since this application expects an application formatted message and will not be able to process the MQSonar echo message.

4.2 Executing the Ping Component

To measure message performance, the measurement starts by running **MQSONAR** at the command prompt. Change to the folder where MQSonar is installed above and run the mqsonar script (mqsonar.cmd or mqsonar.sh). The basic syntax is shown below (full details are in section 4.5, NSQPING Command Line Options):

```
mqsonar QueueName Qmgr_name Options
```

- **Example 1:** Measure message performance every ten seconds on queue manager **QM1** using **SYSTEM.ADMIN.COMMAND.QUEUE**

```
nsqping SYSTEM.ADMIN.COMMAND.QUEUE QM1 -b100 -d10
```

- **Example 2:** Measure message performance every 20 seconds on queue manager **QM2** using **REMOTE.CMD.QUEUE**, perform confirm on arrival (**-ccoa**) and confirm on delivery (**-ccod**) measurements:

```
./nsqping.sh REMOTE.CMD.QUEUE QM2 -ccoa -ccod -b100 -d20
```

4.3 MQSonar Sample Output

The sample below is output from **MQSonar** when executed in interactive mode:

```
mqsonar SYSTEM.ADMIN.COMMAND.QUEUE QM_PROD -ccoa -ccod -b10
```

Pinging `QM_PROD(SYSTEM.ADMIN.COMMAND.QUEUE)` using 36 byte 10(msgs) batch

Statistics for queue QM_PROD(SYSTEM.ADMIN.COMMAND.QUEUE):

```
Summary Performance Indicators :
        MINIMUM_ROUND_TRIP        (0.0020 sec/msg)
        MAXIMUM_ROUND_TRIP        (0.0050 sec/msg)
        AVERAGE_ROUND_TRIP        (0.0014 sec/msg)
        AVERAGE_PROPAGATION_TIME  (0.0008 sec/msg)
        AVERAGE_REFLECTION_TIME   (0.0026 sec/msg)
        MESSAGES_SENT             (10)
        CONFIRMED_EXPIRIES        (0)
        CONFIRMED_DELIVERIES      (10)
        CONFIRMED_ARRIVALS        (10)
        CONFIRMED_EXCEPTIONS      (0)
        REPORTS_RECEIVED          (20)
        RESPONSES_RECEIVED        (10)
        MESSAGES_RECEIVED         (30)
        BYTES_SENT                (360)
        BYTES_RECEIVED            (360)
        RESPONSE_REQUEST_RATIO    (100.0000%)

General Performance Indicators :
        TOTAL_PUT_TIME            (0.0060 sec)
        TOTAL_GET_TIME            (0.0110 sec)
        AVERAGE_PUT_RATE          (1666.6667 msg/sec [60000.00 bytes/sec])
        AVERAGE_GET_RATE          (2727.2727 msg/sec [32727.27 bytes/sec])
        PUT_GET_RATIO             (61.1111% [0.61])

Message Performance Indicators :
        GROSS_ROUND_TRIP_RATE     (2352.9412 msg/sec [42352.94 bytes/sec])
        EFFECTIVE_ROUND_TRIP_RATE(1176.4706 msg/sec)
        CONFIRMATION_OVERHEAD     (50.0000% [0.50])
        AVERAGE_ARRIVAL_RATE      (1000.0000 msg/sec])
        AVERAGE_DELIVERY_RATE     (500.0000 msg/sec])
        AVERAGE_MSG_LATENCY       (2.0000 sec]) WITH QDEPTH(1000)
        MAXIMUM_MSG_LATENCY       (0.0200 sec]) WITH QDEPTH(10)

        TOTAL_BATCH_TIME          (0.0650 sec)
        TEST_COMPLETION_CODE      (0)
```

4.4 MQSonar Statistics

Below is a table of **MQSonar** statistics and their descriptions.

Table 4-1. MQSonar Statistics

Statistic	Description
MINIMUM_ROUND_TRIP	The minimum time for the reply to be returned for the request in seconds.
MAXIMUM_ROUND_TRIP	The maximum time for the reply to be returned for the request in seconds.
AVERAGE_ROUND_TRIP	The average time for the reply to be returned for the request in seconds for all messages sent in a batch.
AVERAGE_PROPAGATION_TIME	Average time in seconds for the message to propagate from the ping component to the echo component and the reply to be sent. For this statistic to be accurate, the time stamps between the servers sending the ping and sending the echo must be synchronized.
AVERAGE_REFLECTION_TIME	Average time in seconds for the message to reflect back from the echo component to the ping component and the reply to be read. For this statistic to be accurate, the time stamps between the servers sending the ping and sending the echo must be synchronized.
MESSAGES_SENT	The total number of messages sent included in this report.
CONFIRMED_EXPIRIES	Number of messages which expired before being delivered to the echo application (if coe specified).
CONFIRMED_DELIVERIES	Number of messages which were delivered to the echo application (if cod specified).
CONFIRMED_ARRIVALS	Number of messages which arrived to the echo application (if coa specified).
CONFIRMED_EXCEPTIONS	Number of messages which resulted in exceptions (if coe specified).
REPORTS_RECEIVED	Total number of confirmation report messages received.
RESPONSES_RECEIVED	Total number of response messages received.
MESSAGES_RECEIVED	Total number of response and report messages received.
BYTES_SENT	Total number of bytes sent.
BYTES_RECEIVED	Total number of bytes received.
RESPONSE_REQUEST_RATIO	The ratio of responses received to requests sent. A value of 100 means that all messages sent received responses.
General Performance Indicators	
TOTAL_PUT_TIME	The total time in seconds to put all of the messages
TOTAL_GET_TIME	The total time in seconds spent waiting for the responses to arrive
AVERAGE_PUT_RATE	The potential messages put rate calculated based on messages processed.
AVERAGE_GET_RATE	The potential messages get rate calculated based on messages processed.
PUT_GET_RATIO	The ratio of put rate to get rate. A value greater than one means that MQSonar could put messages faster than get responses.

Table 4-1. MQSonar Statistics

Statistic	Description
Message Performance Indicators	
`GROSS_ROUND_TRIP_RATE`	Message throughput rate (request + response) / time
`EFFECTIVE_ROUND_TRIP_RATE`	Effective throughput rate (request + response + report) / time
`CONFIRMATION_OVERHEAD`	Percent of messages resulting from report options
`AVERAGE_ARRIVAL_RATE`	Average rate messages arrived at the destination (based on coa messages)
`AVERAGE_DELIVERY_RATE`	Average rate messages arrived at the destination (based on cod messages)
`AVERAGE_MSG_LATENCY`	Average time between arrival and delivery and number of messages on queue
`MAXIMUM_MSG_LATENCY`	Maximum time between arrival and delivery and number of messages on queue
`TOTAL_BATCH_TIME`	The Elapsed time in seconds to process the entire batch.
`TEST_COMPLETION_CODE`	Completion code for the test. 0 indicates that all processing was normal.

When running with report options, the following messages are produced:

```
RESPONSE:  MSGID(1) RND_TRP_MS(0.00) RND_TRP_ON_Q_MS(15.00)
MSG_AGE_MS(1.00) PUT_APPL_NAME(nsqecho.exe) PUT_DATE(20120713)
PUT_TIME(20305078)
```

Response: Produced to show the time that the response message completed the circuit and arrived back at the nsqping application. Verbose option 1.

```
MQFB_COA:  MSGID(1) RND_TRP_MS(0.00) RND_TRP_ON_Q_MS(15.00)
MSG_AGE_MS(1.00) PUT_APPL_NAME(Prod_QMGR1)  PUT_DATE(20120713)
PUT_TIME(20305078)
```

Confirm on Arrival: Produced to show the time that the target message arrived on the target queue. Verbose option 2.

```
MQFB_COD:  MSGID(1) RND_TRP_MS(0.00) RND_TRP_ON_Q_MS(15.00)
MSG_AGE_MS(1.00) PUT_APPL_NAME(Prod_QMGR1) PUT_DATE(20120713)
PUT_TIME(20305078)
```

Confirm on Delivery: Produced to show the time that the target message was consumed from the target queue by the application. The time difference between arrival and delivery is time spent waiting for the application to process the message. Verbose option 2.

4.5 NSQPING Command Line Options

When pinging, MQSonar using the program nsqping. The table below lists the nsqping options and their usage. These options can be specified when running the mqsonar script file.

Table 4-2. NSQPING Command Line Options and Usages

Property	**Description**
`-m<`*`QmgrName`*`>`	Specifies the name of the local queue manager.
`-q<`*`QueueName`*`>`	Queue name of any type (application processing messages put to this queue must not require an application formatted message).
`-b<`*`Batch`*`>`	Number of messages in a batch.
`-s<`*`Bytes`*`>`	Size, in bytes, of generated messages (maximum 32K).
`-d<`*`Interval`*`>`	Interval, in seconds, that ping messages are generated. An interval of 0 (zero) indicates to ping only once.
`-w<`*`Wait`*`>`	Wait interval, in milliseconds, before timeout.
`-e<`*`Expiry`*`>`	Message expires after Expiry time in 0.1 seconds.
`-u<`*`AltUser`*`>`	Issue ping on behalf of another user.
`-ccoa`	Generate confirm-on-arrival report.
`-ccod`	Generate confirm-on-delivery report.
`-cexp`	Generate confirm-on-expiry report.
`-cexc`	Generate confirm-on-exception report.
`-cdlq`	Generate message that will go onto DLQ if undelivered.
`-h<`*`Host`*`>`	Host name of the target AutoPilot M6 Process Wrapper. (See Chapter 5.)
`-p<`*`Port`*`>`	Port number of the target AutoPilot M6 Process Wrapper. (See Chapter 5.)
`-v<0|1|2>`	Set verbose mode to 0 – summary; 1 – reports; 2 – all.

4.6 NSRPL Command Line Options

The mqsonar_echo script uses the nsrpl program. The table below lists the **nsrpl** options which can be used to modify the script executed when running with mqsonar and their usage.

Table 4-3. NSRPL Command Line Options and Usages

Parameters	Description
`QueueName`	Queue name of any type.
`*`	Must be the 2nd parameter for the program.
`QmgrName`	Specifies the name of the local queue manager.
`Waittime`	Time to spend waiting for a message before the application will terminate. In this way, the application can be launched at the start of a test and will terminate automatically when messages are no longer being sent
`NO|YES`	If NO is specified, the response message is exactly the same as the request message. If NO is not specified, the message will have "-reply generated" added to it. This may be useful when running message tracking programs like Nastel TransactionWorks to capture message traffic.

Chapter 5: Integrating MQSonar with AutoPilot M6

MQSonar can be integrated into Nastel's AutoPilot M6 to provide detailed message performance metrics. These metrics can be used to analyze the behavior of the queue manager message performance.

	NOTE:	AutoPilot M6 is shipped separately and not included in the MQSonar distribution.

The following diagram depicts the integration path.

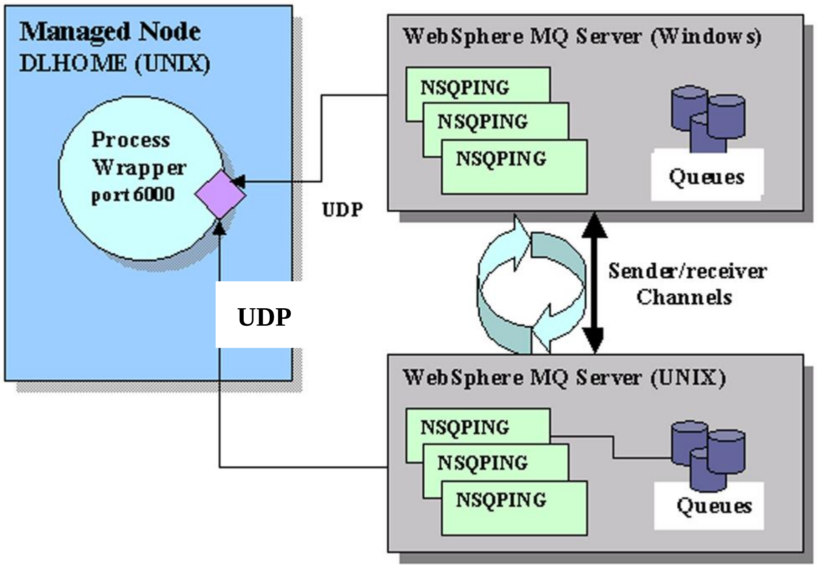


Figure 5-1. Queue MQSonar Integration Path

Integrate as follows:

1. **AutoPilot M6 Configuration:** Deploy an instance of AutoPilot M6 process wrapper on a managed node close or same machine as the source of the metrics. See *AutoPilot M6 Configuration*, section 5.1.
2. **Configuring Process Wrapper Options:** Configure process wrapper to accept WebSphere MQ performance metrics from MQSonar. See *Configuring Process Wrapper*, section 5.1.2.

	NOTE:	Due to unreliable nature of UDP, it is highly recommended that process wrapper be deployed on the same machine as NSQPING. This also means that AutoPilot Server should be running on or in close network proximity to the machine running MQSonar.

Once AutoPilot M6 process wrapper is deployed and configured, do the following:

1. **Measuring Message Performance:** Run MQSonar as a daemon to collect message performance metrics. See *Publishing Messaging Flow Performance*, section 5.2.
2. **Monitor Message Metrics:** Create business view to monitor collected metrics. See *Monitor Message Metrics*, section 5.3.

5.1 AutoPilot M6 Configuration

The following steps are performed to accept WebSphere MQ message performance metrics from MQSonar.

1. **Deploy process wrapper:** Process wrapper is a generic AutoPilot M6 monitor capable of accepting facts from other applications via UDP. It can also start processes, restart processes and record published information into a JDBC database.
2. **Configure process wrapper:** Setting options required to accept information from MQSonar. This step is required to monitor WebSphere MQ message performance metrics.

5.1.1 Deploying Process Wrapper

To deploy and configure an instance of a process wrapper, do the following:

1. Right click on the desired managed node.
2. Select **Deploy Expert** > **Wrappers** > **Process Wrapper**.
3. Configure Process Wrapper for context, name, and to accept fact on the UDP port (6000) that matches the port configured in MQSonar. (See *Configuring Process Wrapper*, section 5.1.2.)
4. Click **Deploy** to deploy the expert on the managed node.

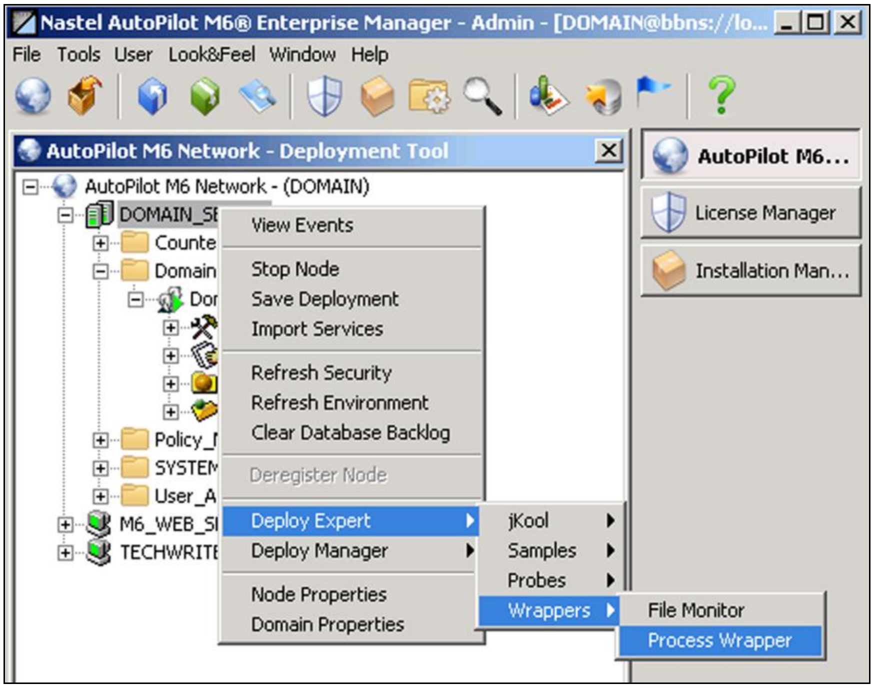
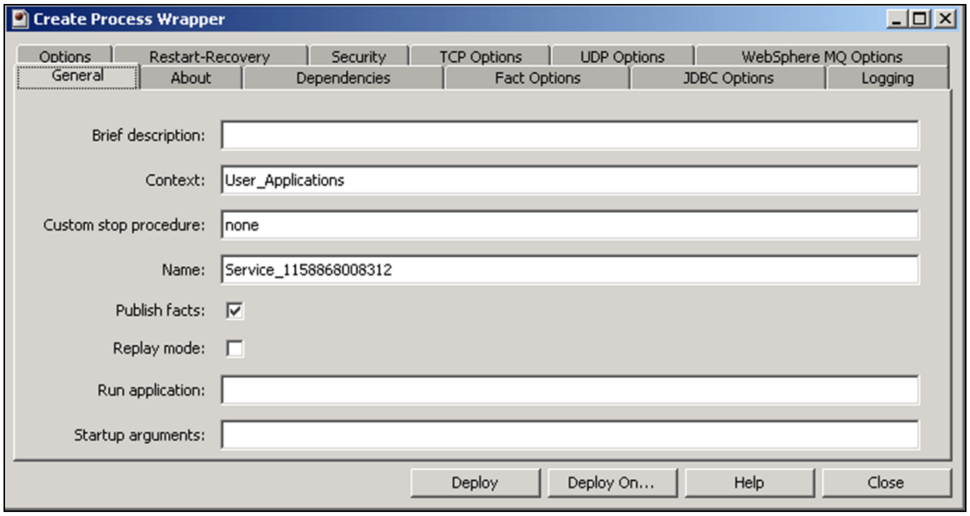


Figure 5-2. Deploying Process Wrappers

Once selected the process wrapper configuration properties will be displayed, as shown below.



Brief description:

Context: User_Applications

Custom stop procedure: none

Name: Service_1158868008312

Publish facts: ☑

Replay mode: ☐

Run application:

Startup arguments:

Figure 5-3. Process Wrapper Configuration Properties

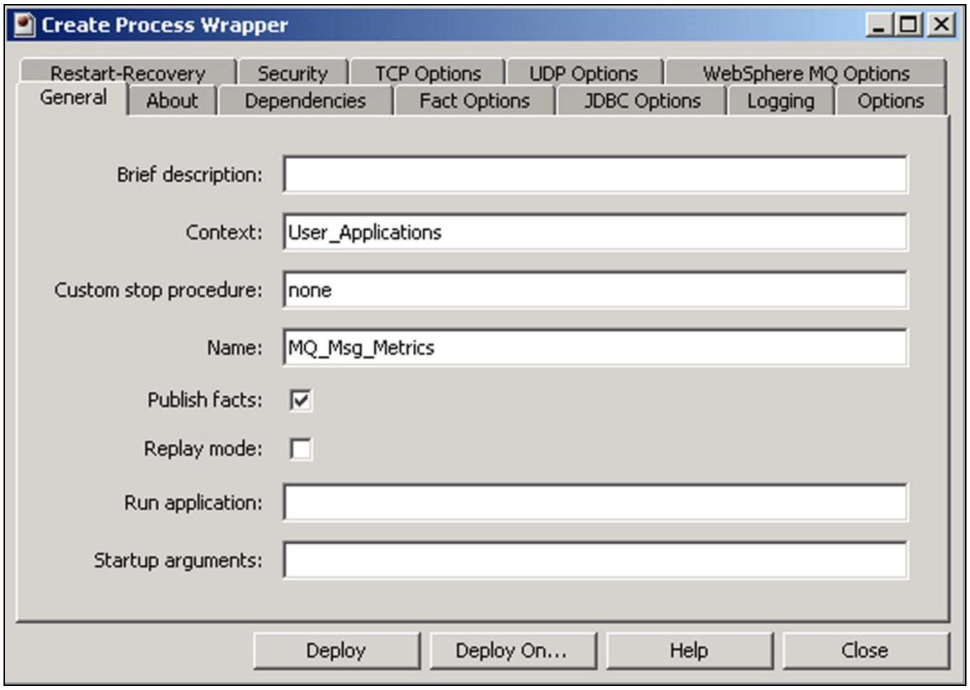
5.1.2 Configuring Process Wrapper

The following process wrapper properties must be configured:

1. **General:** At a minimum *Context* and *Name* must be configured.

Table 5-1. Process Wrapper Expert: General

Property	Description
Brief description	A short, user-defined description of the service.
Context	A user define category that will be registered with the domain server. The default is: User_Applications
Custom stop procedure	Application or script that gracefully shuts down started applications. The default is: none
Name	Name that uniquely identifies the service in the domain. The default name is system assigned with the word “service” and 12 random digits. (Example: service_123456789012)
Publish facts	Check to enable/disable. Publishes received facts locally when enabled.
Replay mode	Check to enable/disable. Emulates fact source and replays all received facts.
Run application	Fully qualified name of application/script/process to run. (Example: `usr/bin/tar`)
Startup arguments	Start–up parameters that are passed to the application. (Example: `cvf mytar/opt/nastel`)



Brief description:

Context: User_Applications

Custom stop procedure: none

Name: MQ_Msg_Metrics

Publish facts: ☑

Replay mode: ☐

Run application:

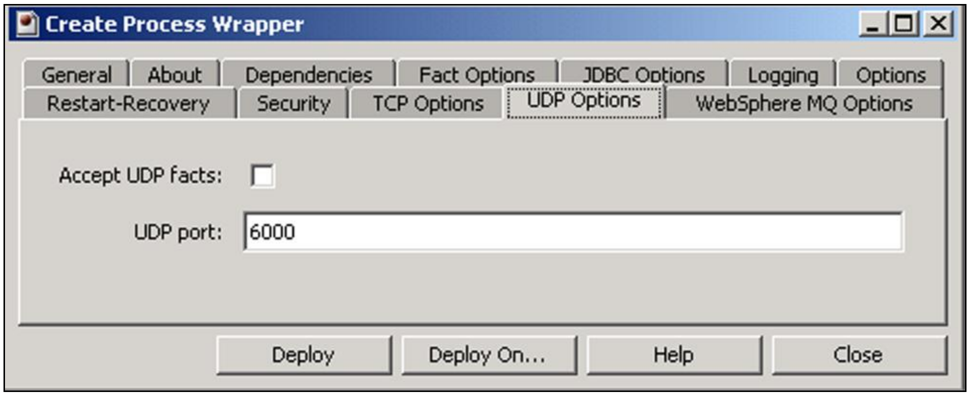
Startup arguments:

Figure 5-4. Process Wrapper General Configuration Properties

2. **UDP Options:** Both properties on this screen must be configured.

Table 5-2. Process Wrapper Expert: UDP Options

Property	Description
Accept UDP facts	Check to enable/disable.
UDP port	Unique port on which process wrapper will accept the incoming performance metrics. Default is 6000.



Accept UDP facts: ☐

UDP port: 6000

Figure 5-5. Process Wrapper UDP Option Configuration Properties

5.2 Publishing Message Flow Performance

To publish message performance metrics using MQSonar, execute the command adding the -h and -p option to match configuration of the process wrapper. Multiple instances of MQSonar can publish performance metrics to the same process wrapper.

```
mqsonar QueueName Qmgr_name –hHostname -pPort
```

- **Example 1:** Measure message performance every 10 seconds on queue manager **QM1** over **SYSTEM.ADMIN.COMMAND.QUEUE** publishing the data to AutoPilot on server localhost at port 6060.

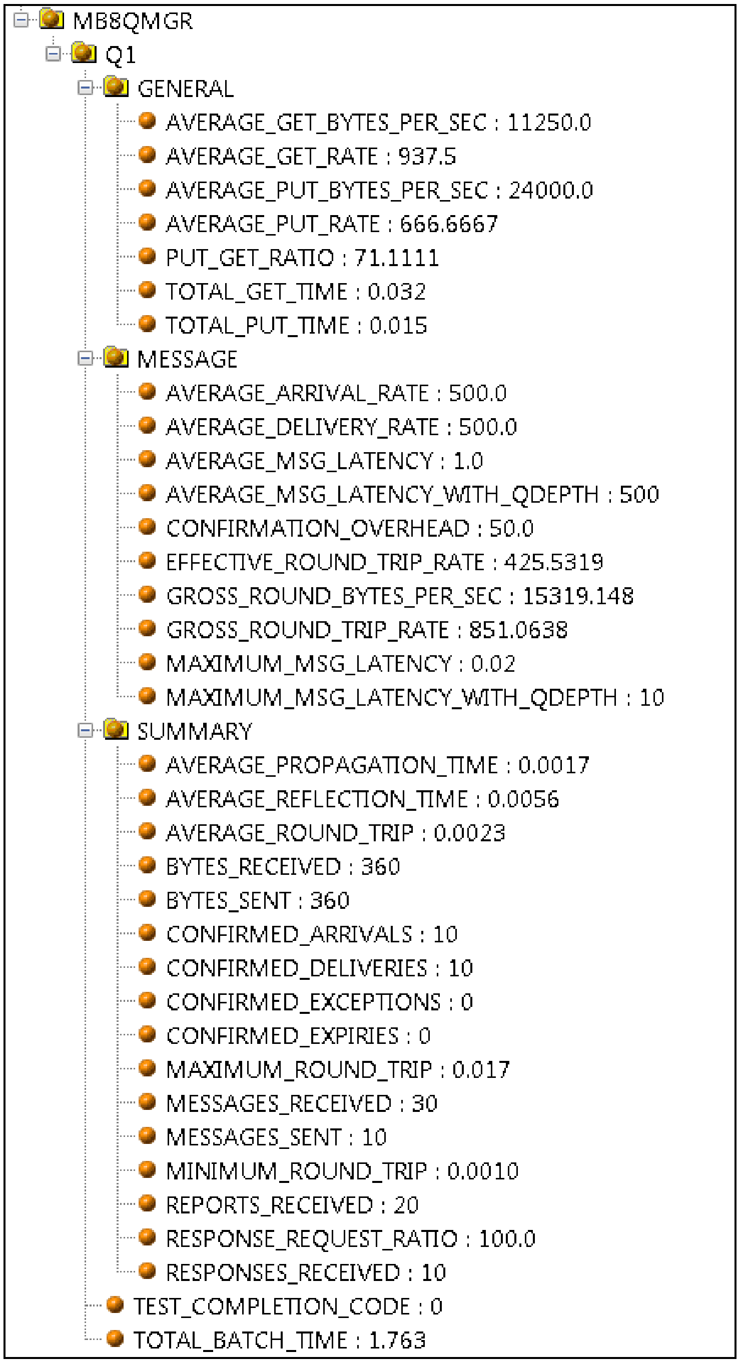
  ```
  mqsonar SYSTEM.ADMIN.COMMAND.QUEUE QM1 –b100 –d10 -hlocalhost -p6060
  ```

- **Example 2**: Measure message performance every 20 seconds on queue manager QM2 over REMOTE.CMD.QUEUE, perform confirm on arrival (-ccoa) and confirm on delivery (-ccod) measurements, publishing the data to AutoPilot on server serverpc at port 6000.

  ```
  ./mqsonar.sh REMOTE.CMD.QUEUE QM2 -ccoa –ccod –b100 –d20 -hserverpc -p6000
  ```

5.3 Monitor Message Metrics

All message performance metrics are available under the process wrapper where they are published by MQSonar, grouped by queue manager and queue, as shown in the figure below.



- MB8QMGR
  - Q1
    - GENERAL
      - AVERAGE_GET_BYTES_PER_SEC : 11250.0
      - AVERAGE_GET_RATE : 937.5
      - AVERAGE_PUT_BYTES_PER_SEC : 24000.0
      - AVERAGE_PUT_RATE : 666.6667
      - PUT_GET_RATIO : 71.1111
      - TOTAL_GET_TIME : 0.032
      - TOTAL_PUT_TIME : 0.015
    - MESSAGE
      - AVERAGE_ARRIVAL_RATE : 500.0
      - AVERAGE_DELIVERY_RATE : 500.0
      - AVERAGE_MSG_LATENCY : 1.0
      - AVERAGE_MSG_LATENCY_WITH_QDEPTH : 500
      - CONFIRMATION_OVERHEAD : 50.0
      - EFFECTIVE_ROUND_TRIP_RATE : 425.5319
      - GROSS_ROUND_BYTES_PER_SEC : 15319.148
      - GROSS_ROUND_TRIP_RATE : 851.0638
      - MAXIMUM_MSG_LATENCY : 0.02
      - MAXIMUM_MSG_LATENCY_WITH_QDEPTH : 10
    - SUMMARY
      - AVERAGE_PROPAGATION_TIME : 0.0017
      - AVERAGE_REFLECTION_TIME : 0.0056
      - AVERAGE_ROUND_TRIP : 0.0023
      - BYTES_RECEIVED : 360
      - BYTES_SENT : 360
      - CONFIRMED_ARRIVALS : 10
      - CONFIRMED_DELIVERIES : 10
      - CONFIRMED_EXCEPTIONS : 0
      - CONFIRMED_EXPIRIES : 0
      - MAXIMUM_ROUND_TRIP : 0.017
      - MESSAGES_RECEIVED : 30
      - MESSAGES_SENT : 10
      - MINIMUM_ROUND_TRIP : 0.0010
      - REPORTS_RECEIVED : 20
      - RESPONSE_REQUEST_RATIO : 100.0
      - RESPONSES_RECEIVED : 10
    - TEST_COMPLETION_CODE : 0
    - TOTAL_BATCH_TIME : 1.763

Figure 5-6. Published Message Performance Metrics

All metrics published by MQSonar can be included in user defined business views with rules, automation, alerts and notifications.

	NOTE:	Refer to *AutoPilot M6 User's Guide* for more information about business views.

Chapter 6: Troubleshooting

Additional troubleshooting information is available in the community forum. To access the forum, use the link on the main page or you can link directly using http://www.nastel.com/support/mqsonar/.

Issue: When using the command server as the echo component, nsqping returns the message "Pinging queue manager completed with RC(2033)"

What to do:

Verify that the queue specified ends with a SYSTEM.ADMIN.COMMAND.QUEUE.

If using multiple queue managers, verify that the channels are started in both directions.

Verify that the user running the MQSonar is defined and is authorized to access MQ, the queue specified, and any queues in the path of the MQSonar route.

Include the -ccoa option and execute with -v2 to see if the messages are arriving on the target queue.

If using the WMQ command server as the echo component, the maximum message size is 32K.

Issue: When running either command, get the message "error while loading shared libraries: libmqm_r.so" or similar.

What to do:

This indicates that the WebSphere MQ libraries are not set. Depending on the release of MQ, there are various options. You can also modify the script to specify the correct location.

This page intentionally left blank.

Index

This page intentionally left blank.