EMSTools – User’s Guide

Version 1.3.7

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| 1.3.5 | June 07, 2017 | Pierre Ayel | Updated for release 1.3.5 of EMSTools |
| 1.3.7 | February 15, 2018 | Pierre Ayel | Updated for release 1.3.7 of EMSTools:   * Relaxed input message format, * Server, user and password parameters are now optional |
|  |  |  |  |

Copyright Notice

COPYRIGHT© 2016 TIBCO Software Inc. This document is unpublished and the foregoing notice is affixed to protect TIBCO Software Inc. in the event of inadvertent publication. All rights reserved. No part of this document may be reproduced in any form, including photocopying or transmission electronically to any computer, without prior written consent of TIBCO Software Inc. The information contained in this document is confidential and proprietary to TIBCO Software Inc. and may not be used or disclosed except as expressly authorized in writing by TIBCO Software Inc. Copyright protection includes material generated from our software programs displayed on the screen, such as icons, screen displays, and the like.

Trademarks

All brand and product names are trademarks or registered trademarks of their respective holders and are hereby acknowledged. Technologies described herein are either covered by existing patents or patent applications are in progress.

Confidentiality

The information in this document is subject to change without notice. This document contains information that is confidential and proprietary to TIBCO Software Inc. and its affiliates and may not be copied, published, or disclosed to others, or used for any purposes other than review, without written authorization of an officer of TIBCO Software Inc. Submission of this document does not represent a commitment to implement any portion of this specification in the products of the submitters.

Content Warranty

The information in this document is subject to change without notice. THIS DOCUMENT IS PROVIDED "AS IS" AND TIBCO MAKES NO WARRANTY, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO ALL WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TIBCO Software Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Export

This document and related technical data, are subject to U.S. export control laws, including without limitation the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations of other countries. You agree not to export or re-export this document in any form in violation of the applicable export or import laws of the United States or any foreign jurisdiction.

For more information, please contact:

TIBCO Software Inc.  
3303 Hillview Avenue  
Palo Alto, CA 94304  
USA

**Table of Contents**

1 Introduction 6

1.1 Overview 6

1.2 Audience 6

2 Installation 7

2.1 Software Location 7

2.2 Installation on Unix 7

2.3 Installation on Windows 7

3 Common Options 8

3.1 <server-options> 8

3.2 <ssl-options> 10

3.3 <other-options> 11

4 File Formats 14

4.1 General Format 14

4.2 Received Messages 14

4.3 Sent Messages 15

5 EMSQueueBrowser 17

5.1 Description 17

5.2 Usage 17

6 EMSQueueReceiver 19

6.1 Description 19

6.2 Usage 19

6.3 Request-Reply 21

7 EMSQueueSender 22

7.1 Description 22

7.2 Usage 22

7.3 Sending Loop 23

8 EMSTopicListener 25

8.1 Description 25

8.2 Usage 25

9 EMSTopicPublisher 27

9.1 Description 27

9.2 Usage 27

9.3 Sending Loop 28

10 EMSTestQueueConnection 30

10.1 Description 30

10.2 Usage 30

11 EMSTestTopicConnection 31

11.1 Description 31

11.2 Usage 31

12 EMSAdmin 32

12.1 Description 32

12.2 Usage 32

13 EMSAdmin-all 34

13.1 Description 34

13.2 Usage 34

14 Frequently Asked Questions 35

14.1 How to specify TIBCO JNDI reconnection parameters? 35

14.2 How to connect to a Solace Appliance? 35

**Table of Figures**

**No table of figures entries found.**

**Table of Tables**

Table 1: <server-options> 8

Table 2: <ssl-options> 10

Table 3: <other-options> 12

Table 4: EMSQueueBrowser <options> 17

Table 5: EMSQueueReceiver <options> 20

Table 6: EMSAdmin-all <server-options> 22

Table 7: EMSTopicListener-<options> 25

Table 8: EMSTopicPublisher <options> 27

Table 9: EMSTestQueueConnection <options> 30

Table 10: EMSTestTopicConnection <options> 31

Table 11: EMSAdmin <server-options> 32

Table 12: EMSAdmin-all <server-options> 34

1. Introduction

This document is a user’s guide for the EMSTools, a set of command line tools to interact with a TIBCO EMS server. The tools are not a TIBCO product, but a project-ware provided with samples, source code and documentation.

* 1. Overview

EMSTools allows you:

* Publishing messages into a JMS Topic,
* Publishing messages into a JMS Queue,
* Receiving messages from a JMS Topic, and optionally save them into files,
* Consuming messages from a JMS Queue, and optionally save them into files,
* Browsing messages from a JMS Queue, and optionally save them into files,
* Testing the connection parameters to an EMS Server,
* Collecting EMS server data into .CSV files.
  1. Audience

The EMSTools can be used for testing or troubleshooting JMS issues, backing up messages before a purge or even in integration patterns for non-TIBCO systems to publish or receive JMS messages. Consequently, support teams, developers, testers, designers and architects are recommended to read this document.

1. Installation
   1. Software Location

EMSTools are provided in a tar file and a zip file called EMSTools-<version>.tar or EMSTools-<version>.zip respectively.

* 1. Installation on Unix
     1. Pre-Requisites
* A Java runtime environment (JRE) or JDK must be available. If the host has TIBCO BusinessWorks, you can use the JRE installed with it.
* You DO NOT need to install TIBCO EMS client libraries. EMSTools installation contains its own set of TIBCO EMS client libraries.
  + 1. Installation
* Connect as “tibco” user to the machine.
* Extract the archive into the folder /opt/tibco/tools (and create it if it does not exist).

cd /opt/tibco

mkdir tools

cd tools

tar -xvf EMSTools-1\_0\_1.tar

* Change the execution rights of the installed scripts:

chmod u+x EMSTools/bin/\*.sh

* 1. Installation on Windows
     1. Pre-Requisites
* A Java runtime environment (JRE) or JDK must be available. If the host has TIBCO BusinessWorks, you can use the JRE installed with it.
* You DO NOT need to install TIBCO EMS client libraries. EMSTools installation contains its own set of TIBCO EMS client libraries.
  + 1. Installation
* Unzip the EMSTools zip file in a folder of your choice.

1. Common Options

EMSTools are a set of command line scripts for Windows and UNIX

This chapter describes the command line options which are common to all scripts.

* 1. <server-options>

The following command line parameters must be provided for the script to connect to one EMS server:

When using a direct TCP or SSL connection:

-server <url>

-user <user>

[[-password <password>]

[-password-file <password-file>]]

[-clientid <clientid>]

[-factory-clientid <factory-clientid>]

When using a JNDI connection:

-provider <jdni-url> or –jndi\_url <jndi-url>

-user <user>

[[-password <password>]

[-password-file <password-file>]]

-factory <factory>

[-clientid <clientid>]

[-factory-clientid <factory-clientid>]

[-jndi\_provider\_factory <jndi-class>]

Options can be provided in any order.

Options are listed in this table:

Table 1: <server-options>

| Parameter | Description | Examples |
| --- | --- | --- |
| <url> | The URL of the server.  -server <url> is optional since version 1.3.7 (default value is tcp://localhost:7222) | tcp://localhost:2222  tcp://primary:1234,tcp://secondary:1234  ssl://host1:4567,ssl://host2:3456 |
| <user> | The username/login to the server.  When using a JNDI connection, the login is used for both the first JNDI connection and the second connection using the connection factories details.  -user <user> is optional since version 1.3.7 (default value is admin | esb |
| <password> | The password to the server.  When using a JNDI connection, the password is used for both the first JNDI connection and the second connection using the connection factories details  -password <password> is optional since version 1.3.7 (default value is empty string) | esb |
| <password-file> | The name of a text file containing the password.  This allows hiding the actual password from the command line in case you may use one EMSTools script for a long time. This will avoid other users finding the password by searching the list of processes running on the machine. | c:/temp/mypassword.txt |
| <jndi-url> | The JNDI URL of the server.  This option is supported since version 1.3.3 | tibjmsnaming://localhost:222  tibjmsnaming://primary:1234,…  …tibjmsnaming://secondary:1234 |
| <factory> | The name of the queue connection factory or topic connection factory.  -server <url> is optional since version 1.3.7 (default value is tcp://localhost:7222) | MyQueueConnectionFactory |
| <clientid> | Optional:  A client id (user-chosen identifier) to setup on the connection.  If another connection is already using the same client id the EMS server will set your to <empty> or may throw an error which can be ignored. | Pierre |
| <factory-clientid> | Optional:  A client id (user-chosen identifier) to setup on the JNDI connection.  If another connection is already using the same client id the EMS server will set your to <empty> or may throw an error which can be ignored. | Pierre |
| <jndi-class> | Optional:  The Java class name of the JNDI provider factory. If missing, the default class is used: com.tibco.tibjms.naming.TibjmsInitialContextFactory |  |

***Notes:***

* When using a JNDI connection, the JDNI factory Java class name is set to “com.tibco.tibjms.naming.TibjmsInitialContextFactory” by default. Since version 1.3.3, you can provide the class-name in the –jndi\_provider\_factory option.
  1. <ssl-options>

If you connect to one EMS server via SSL, you must provide the following command line options:

-ssl

-ssl-verify\_host

-ssl-verify-hostname

-ssl-hostname <ssl\_expected\_hostname>

[-ssl-trusted <trusted\_cert\_file1> [-ssl-trusted <trusted\_cert\_file2>] …]

-ssl-identity <ssl\_identity>

-ssl-key <ssl\_key>

-ssl-password <ssl\_password>

-ssl-vendor <ssl\_vendor>

-ssl-ciphers <ssl\_ciphers>

-ssl-trace

-ssl-debug-trace

Options are as follow (all are optional, the default value is not documented, if you are not sure of what default value is used, specify the option on the command line):

Table 2: <ssl-options>

| Parameter | Description | Examples |
| --- | --- | --- |
| -ssl | Enables SSL mode |  |
| -ssl\_vertify\_host | Indicates that the certificate sent by the EMS server must be verified.  If this option is missing, the certificate sent by the EMS server is trusted and not verified. |  |
| -ssl\_verify\_hostname | Indicates that the hostname in the certificate send by the EMS server must match the hostname from the server URL or <ssl\_expected\_hostname> if specified.  If this option is missing, the hostname inside certificate sent by the EMS server is trusted and not verified. |  |
| <ssl\_expected\_hostname> | The hostname expected in the certificate send by the EMS server.  This option is ignored if the option “-ssl\_verify\_hostname” is missing. |  |
| <trusted\_cert\_file1>  <trusted\_cert\_file2>  … | A set of trusted certificate files that will be matched against the certificate sent by the EMS server is the ”-ssl\_verify\_host” option is present. |  |
| <ssl\_identity> | If the EMS server expects clients to present their own certificates, those options will allows EMSTools to use the certificate present in a identity file |  |
| <ssl\_key> |  |
| <ssl\_password> |  |
| <ssl\_vendor> | TBC |  |
| <ssl\_ciphers> | TBC |  |
| -ssl\_trace | Enables SSL tracing. |  |
| -ssl\_debug\_trace | Enables SSL debug tracing. |  |

* 1. <other-options>

You can also provide the following command line options:

-debug

-noUnmap

-close-in-callback <close\_in\_callback>

-connect-attempts <connect\_attempts>

-connect-attempt-timeout <connect\_attemps\_timeout>

-daemon[-\_]dispatcher <daemon\_dispatcher>

-trace-ft-events <trace\_ft\_events>

-trace-ft-switch <trace\_ft\_switch>

-message-encoding <message\_encoding>

-multicast-daemon <multicast\_daemon>

-multicast-enabled <multicast\_enabled>

-ping-interval <ping\_interval>

-reconnect-attempts <reconnect\_attempts>

-reconnect-attempt-timeout <reconnect\_attempt\_timeout>

-socket-connect-timeout <socke\_connect\_timeout>

-socket-receive <socket\_received>

-socket-send <socket\_send>

-trace-file <trace\_file>

-quiet

-log4j

-jndi\_property <jdniname=value>

Options are listed in this table (all are optional):

Table 3: <other-options>

| Parameter | Description | Examples |
| --- | --- | --- |
| -debug | Enables debug tracing. |  |
| -noUnmap | Disables extracting the “message\_bytes” entry from received MapMessages and using it as output message (see “4.2 Received Messages” section for more information). |  |
| <close\_in\_callback> | Those options change the value of Java system properties, which name is the corresponding PROP\_XXX contant defined in the Java class com.tibco.tibjms.Tibjms (see TIBCO EMS Java API reference for more information). | -connect\_attempts 3  Will force EMSTools first connection to be retried 3 times. |
| <connect\_attempts> |
| <connect\_attemps\_timeout> |
| <daemon\_dispatcher> |
| <trace\_ft\_events> |
| <trace\_ft\_switch> |
| <message\_encoding> |
| <multicast\_daemon> |
| <multicast\_enabled> |
| <ping\_interval> |
| <reconnect\_attempts> |
| <reconnect\_attempt\_timeout> |
| <socke\_connect\_timeout> |
| <socket\_received> |
| <socket\_send> |
| <trace\_file> |
| -quiet | Suppress some of logging information (only reports errors or issues). |  |
| -log4j | Enables using LOG4J for logging instead of standard output. |  |
| <jndiname=value> | Property name and value to add into the JNDI initial context before a connection is made to JNDI.  This option is supported since version 1.3.3. | -jndi\_property TBC=myvpn |

1. File Formats
   1. General Format

When EMSTools writes or reads one message, the text written or read from the text files is as following:

$Msg$

$MsgHeader$

<name>=<value>

<name>=<value>

…

$MsgProperties$

<name>=<value>

<name>=<value>

…

$MsgBody$

…

$MsgEnd$

$Msg$ is optional. EMSQueueReceiver writes $MsgRequest$ and $MsgResponse$ instead when working in request-reply mode, or $MsgNotify$ when working in default mode.

$MsgHeader$ indicates the next <name>=<value> lines contain message JMS properties such as Message ID, Correlation ID, etc…

Since version 1.3.7, $MsgHeader$ is optional.

$MsgProperties$ indicates the next <name>=<value> lines contain user messages properties. Note that the type of the property is not indicated.

Since version 1.3.7, $MsgProperties$ is optional.

$MsgBody$ indicates the next line until $MsgEnd$ contain the payload of the message.

Since version 1.3.7, if the file does not contain any line starting with $, the whole message content is used for the message payload.

* 1. Received Messages

When a message is received, all properties and details of the message are written into the file.

***Example:***

$Msg$

$MsgHeader$

JMSMessageID=ID:EMSBIZ-LOC1.1EAC5280FF532B:5

JMSTimestamp=Mon Nov 11 16:09:51 GMT 2013

JMSDestination=Queue[TSL.UK.DEV.test.XML.V1\_0]

JMSDeliveryMode=PERSISTENT

JMSPriority=4

$MsgProperties$

JMSXDeliveryCount=2

EVENT\_NAME=xxx

$MsgBody$

<?xml version="1.0" encoding="UTF-8"?>

<MESSAGE ID="ADJ">&#xD;

<RECORD STORENO="6997" PRODUCTCODE="58142751" MESSAGEREFERENCENUMBER="32783" EFFECTIVEDATETIME="20131109235900" RETURNCODE="0"/>&#xD;

</MESSAGE>

$MsgEnd$

If a received message is a MapMessage (instead of a TextMessage), EMSTools will automatically extract the payload contains in the map entry called “message\_bytes” and write it into the output file instead of the original received MapMessage. You can disable this feature by providing the command line option “-noUnmap”.

* 1. Sent Messages

When you send a message, you do not need to provide all JMS properties. Even if you do so, some will be ignored.

***Example:***

$Msg$

$MsgHeader$

JMSDeliveryMode=PERSISTENT

JMSPriority=4

$MsgProperties$

EVENT\_NAME=xxx

$MsgBody$

<?xml version="1.0" encoding="UTF-8"?>

<MESSAGE ID="ADJ">&#xD;

<RECORD STORENO="6997" PRODUCTCODE="58142751" MESSAGEREFERENCENUMBER="32783" EFFECTIVEDATETIME="20131109235900" RETURNCODE="0"/>&#xD;

</MESSAGE>

$MsgEnd$

The minimum you can provide is as following:

$MsgHeader$

$MsgBody$

<?xml version="1.0" encoding="UTF-8"?>

<MESSAGE ID="ADJ">&#xD;

<RECORD STORENO="6997" PRODUCTCODE="58142751" MESSAGEREFERENCENUMBER="32783" EFFECTIVEDATETIME="20131109235900" RETURNCODE="0"/>&#xD;

</MESSAGE>

$MsgEnd$

For JMSDeliveryMode, you can write “PERSISTENT” or “NON\_PERSISTENT”.

For user properties, if the value is “true” or “false” EMSTools treat the property as a Boolean property, otherwise EMSTools treat the property as a String property by default.

EMSTools always add the Boolean property JMS\_TIBCO\_PRESERVE\_UNDELIVERED with value “true” to all sent messages.

If the file did not specify any JMSCorrelationID, EMSTools will always add a random one (except for EMSQueueSender reply messages if the received message already contains one).

1. EMSQueueBrowser

EMSTools are a set of command line scripts for Windows and Unix. Each script usage is documented in its own chapter.

* 1. Description

**EMSQueueBrowser** allows browsing one JMS Queue, e.g. reading the messages but keeping them in the JMS Queue. Messages that are currently being consumed by existing consumers will not be read.

**EMSQueueBrowser** writes all the read messages into standard output and can also write them into a single text file or write one text file per message.

**EMSQueueBrowser** can then be used to back up the content of a JMS queue before messages are purged for example.

* 1. Usage

From any working directory, type the following:

<install-folder>/EMSTools/bin/EMSQueueBrowser.sh

<server-options>

<ssl-options>

<other-options>

-queue <queue> or –jndi\_queue <jndi-queue>

[-log <output-file>]

[-count <count>]

[-selector <selector>]

Or the following command (on Windows):

<install-folder>\EMSTools\bin\EMSQueueBrowser.bat

<server-options>

<ssl-options>

<other-options>

-queue <queue> or –jndi\_queue <jndi-queue>

[-log <output-file>]

[-count <count>]

[-selector <selector>]

Options are listed in this table:

Table 4: EMSQueueBrowser <options>

| Parameter | Description | Examples |
| --- | --- | --- |
| <server-options> | Common Options to connect to the EMS Server (see chapter “3 Common Options” for more information). |  |
| <ssl-options> |  |
| <other-options> |  |
| <queue> | The name of the JMS queue (the full name). | TSL.UK.DEV.Test.XML.V1\_0 |
| <jndi-queue> | The JNDI name of the JMS queue.  The connection must have been established with the server JNDI interface.  Since version 1.3.3 | Test.XML.V1\_0 |
| <output-file> | Optional:  The name of a file where all read messages are written. If the name is the name of an existing folder, each message is written in its own file and the location is <output-file>/Queue[<queue>]/<msg-id>.msg | /tmp/mymessage-folder  /tmp/mymessage.txt |
| <count> | Optional:  The maximum number of messages to read. If you do not provide this option, all messages that the tool can read are read without limit | 3 |
| <selector> | Optional:  A message selector to filter messages to read.  Caution: using message selector in case of backlog can degrade the performance of the EMS server. | EVENT\_NAME=’xxx’ |

1. EMSQueueReceiver

EMSTools are a set of command line scripts for Windows and Unix. Each script usage is documented in its own chapter.

* 1. Description

**EMSQueueReceiver** allows consuming messages from one JMS Queue.

**EMSQueueReceiver** writes all the consumed messages into standard output and can also write them into a single text file or write one text file per message.

**EMSQueueReceiver** is using the AUTO\_ACKNOWLEDGE mode so all messages received by the tools are automatically removed from the JMS queue (even before writing them in the output file or folder may fail).

* 1. Usage

From any working directory, type the following:

<install-folder>/EMSTools/bin/EMSQueueReceiver.sh

<server-options>

<ssl-options>

<other-options>

-queue <queue> or –jndi\_queue <jndi-queue>

[-log <output-file>]

[-count <count>]

[-selector <selector>]

[-timeout <timeout>]

[-infile <infile1> [-infile <infile2>]…]

[-noreply]

Or the following command (on Windows):

<install-folder>\EMSTools\bin\EMSQueueReceiver.bat

<server-options>

<ssl-options>

<other-options>

-queue <queue> or –jndi\_queue <jndi-queue>

[-log <output-file>]

[-count <count>]

[-selector <selector>]

[-timeout <timeout>]

[-infile <infile1> [-infile <infile2>]…]

[-noreply]

Options are listed in this table:

Table 5: EMSQueueReceiver <options>

| Parameter | Description | Examples |
| --- | --- | --- |
| <server-options> | Common Options to connect to the EMS Server (see chapter “3 Common Options” for more information). |  |
| <ssl-options> |  |
| <other-options> |  |
| <queue> | The name of the JMS queue. | TSL.UK.DEV.Test.XML.V1\_0 |
| <jndi-queue> | The JNDI name of the JMS queue.  The connection must have been established with the server JNDI interface.  Since version 1.3.3 | Test.XML.V1\_0 |
| <output-file> | Optional:  The name of a file where all consumed messages are written. If the name is the name of an existing folder, each message is written in its own file and the location is <output-file>/Queue[<queue>]/<msg-id>.msg . | /tmp/mymessages-folder  /tmp/mymessage.msg |
| <count> | Optional:  The maximum number of messages to read. If you do not provider this option, all messages that the tool can read are read without limit | 3 |
| <selector> | Optional:  A message selector to filter messages to consume.  Caution: using message selector in case of backlog can degrade the performance of the EMS server. | EVENT\_NAME=’xxx’ |
| <timeout> | Optional:  Time to wait for the next message in seconds. When no message is received after the time elapses the script stops. If you do not provider this option, the script consumes messages forever | 30 |
| <infile1>  <infile2>  … | Optional:  In case the received messages has JMSReplyTo property set, this indicates where to read the payload of messages to send as reply. | /tmp/mymessage.msg |
| -noreply | Optional:  Disables the request-reply mode. |  |

* 1. Request-Reply

**EMSQueueReceiver** can act as a request-reply server and publishes a reply messages for each received message if the message has the “JMSReplyTo” property set.

In this case, if you provided one or more <infile1> options, the content of the reply messages is read from the files, otherwise the default payload is the text “message”.

If you provided multiple <infile1> options, each file is used for each received message in a loop.

***Example:***

-infile 1.msg –infile 2.msg

*When the first message arrives, the reply payload is read from the first message in the file 1.msg.*

*When the second message arrives, the reply payload is read from the first message in the file 2.msg.*

*When the third message arrives, the reply payload is read from the first message in the file 1.msg.*

*Etc…*

The reply message is published with properties copied from the received message as following:

Table 1: EMSQueueReceiver reply message properties

| Property | Value |
| --- | --- |
| JMSCorrelationID | Copied from received message. |
| JMSDeliveryMode | Copied from received message.  Set to NON\_PERSISTENT if the name of JMSReplyTo destination starts with “$TMP” (which means it is a temporary topic or temporary queue). |
| JMS\_TIBCO\_PRESERVE\_UNDELIVERED | “true” |
| Operation | Those custom string properties are copied from the received message. |
| Source |
| TransactionId |
| ContractVersion |

1. EMSQueueSender

EMSTools are a set of command line scripts for Windows and Unix. Each script usage is documented in its own chapter.

* 1. Description

**EMSQueueSender** allows sending one or more messages into one JMS Queue.

* 1. Usage

From any working directory, type the following:

<install-folder>/EMSTools/bin/EMSQueueSender.sh

<server-options>

<ssl-options>

<other-options>

-queue <queue> or –jndi\_queue <jndi-queue>

[-count <count>]

[-log <logfile>]

[-delay <delay>]

Or the following command (on Windows):

<install-folder>\EMSTools\bin\EMSQueueSender.bat

<server-options>

<ssl-options>

<other-options>

-queue <queue>

[-count <count>]

[-log <logfile>]

[-delay <delay>]

Options are listed in this table:

Table 6: EMSAdmin-all <server-options>

| Parameter | Description | Examples |
| --- | --- | --- |
| <server-options> | Common Options to connect to the EMS Server (see chapter “3 Common Options” for more information). |  |
| <ssl-options> |  |
| <other-options> |  |
| <queue> | The name of the JMS queue. | TSL.UK.DEV.Test.XML.V1\_0 |
| <jndi-queue> | The JNDI name of the JMS queue.  The connection must have been established with the server JNDI interface.  Since version 1.3.3 | Test.XML.V1\_0 |
| <count> | Optional:  The number of time messages are sent. If you do not provide this option, the default value is 1.  See next paragraphs for more explanations. | 3 |
| <infile1>  <infile2>  … | A set of input files or folders where messages to sent are read from. | /tmp/mymessages/…  Queue[TSL.UK.DEV.Test… |
| <logfile> | Optional:  The name of a file where all sent messages are written. If the name is the name of an existing folder, each message is written in its own file and the location is <output-file>/Queue[<queue>]/<msg-id>.msg | /tmp/mymessages-folder  /tmp/mymessage.msg |
| <delay> | Optional:  The delay in milliseconds between each message publication. If you do not provide this option, the default is 1000ms (1s) | 2000 |

***Notes:***

* In you do provide any <infileX> option, the script sends a default text message in NON\_PERSISTENT mode. The payload of the message is “message<N>” where <N> varies from 1 to <count>.
  1. Sending Loop

If you provide a single “-infile <infile>” option and <infile> is the name of an existing folder, the script will:

* For i = 1 to <count>
  + For each file present in the folder which name ends with “.msg”, read the first message of the file and send it.

If you provide a single “-infile <infile>” option and <infile> is the name of an existing file, the script will:

* For i = 1 to <count>
  + Read the first message of the file and send it.

If you provide multiple “-infile <infile>” options all pointing to existing files, the script will:

* For i = 1 to <count>
  + Read the first message of the first file and send it. The next time this step is done, the next file is read and so on in a loop.

With this mode, you can send semi-random messages.

***Example:***

…/EMSQueueSender.sh … -infile 1.msg –infile 2.msg –infile 3.msg –count 4

*The script will send the message in this fashion:*

*1.msg*

*2.msg*

*3.msg*

*1.msg*

If you provide multiple “-infile <infile>” options pointing to existing files and folders, the script will fail when reading the folders.

1. EMSTopicListener

EMSTools are a set of command line scripts for Windows and Unix. Each script usage is documented in its own chapter.

* 1. Description

**EMSTopicListener** allows receiving messages from one JMS Topic.

**EMSTopicListener** writes all the read messages into standard output and can also write them into a single text file or write one text file per message.

* 1. Usage

From any working directory, type the following:

<install-folder>/EMSTools/bin/EMSTopicListener.sh

<server-options>

<ssl-options>

<other-options>

-topic <topic> or –jndi\_topic <jndi-topic>

[-log <output-file>]

[-count <count>]

[-selector <selector>]

Or the following command (on Windows):

<install-folder>\EMSTools\bin\EMSTopicListener.bat

<server-options>

<ssl-options>

<other-options>

-topic <topic> or –jndi\_topic <jndi-topic>

[-log <output-file>]

[-count <count>]

[-selector <selector>]

Options are listed in this table:

Table 7: EMSTopicListener-<options>

| Parameter | Description | Examples |
| --- | --- | --- |
| <server-options> | Common Options to connect to the EMS Server (see chapter “3 Common Options” for more information). |  |
| <ssl-options> |  |
| <other-options> |  |
| <topic> | The full name of the JMS topic. | TSL.UK.DEV.Test.XML.V1\_0 |
| <jndi-topic> | The JNDI name of the JMS topic.  The connection must have been established with the server JNDI interface.  Since version 1.3.3 | Test.XML.V1\_0 |
| <output-file> | Optional:  The name of a file where all read messages are written. If the name is the name of an existing folder, each message is written in its own file and the location is <output-file>/Queue[<queue>]/<msg-id>.msg | /tmp/mymessages-folder  /tmp/mymessage.msg |
| <count> | Optional:  The maximum number of messages to read. If missing, all messages that the tool can read are read without limit. | 3 |
| <selector> | Optional:  A message selector to filter messages to read.  Caution: using message selector in case of backlog can degrade the performance of the EMS server. | EVENT\_NAME=’xxx’ |

1. EMSTopicPublisher

EMSTools are a set of command line scripts for Windows and Unix. Each script usage is documented in its own chapter.

* 1. Description

**EMSTopicPublisher** allows publishing one or more messages into one JMS Topic.

* 1. Usage

From any working directory, type the following command:

<install-folder>/EMSTools/bin/EMSTopicPublisher.sh

<server-options>

<ssl-options>

<other-options>

-topic <topic> or –jndi\_topic <jndi-topic>

[-count <count>]

[-infile <infile1> [-infile <infile2>] …]

[-log <logfile>]

[-delay <delay>]

Or the following command (on Windows):

<install-folder>\EMSTools\bin\EMSTopicPublisher.bat

<server-options>

<ssl-options>

<other-options>

-topic <topic> or –jndi\_topic <jndi-topic>

[-count <count>]

[-infile <infile1> [-infile <infile2>] …]

[-log <logfile>]

[-delay <delay>]

Options are listed in this table:

Table 8: EMSTopicPublisher <options>

| Parameter | Description | Examples |
| --- | --- | --- |
| <server-options> | Common Options to connect to the EMS Server (see chapter “3 Common Options” for more information). |  |
| <ssl-options> |  |
| <other-options> |  |
| <topic> | The name of the JMS topic. | TSL.UK.DEV.Test.XML.V1\_0 |
| <jndi-topic> | The JNDI name of the JMS topic.  The connection must have been established with the server JNDI interface.  Since version 1.3.3 | Test.XML.V1\_0 |
| <count> | Optional:  The number of time messages are sent. If you do not provide this option, the default value is 1.  See next paragraphs for more explanations. | 3 |
| <infile1>  <infile2>  … | A set of input files or folders where messages to sent are read from. | /tmp/mymessages/…  Queue[TSL.UK.DEV.Test… |
| <logfile> | Optional:  The name of a file where all sent messages are written. If the name is the name of an existing folder, each message is written in its own file and the location is <output-file>/Queue[<queue>]/<msg-id>.msg | /tmp/mymessages-folder  /tmp/mymessage.msg |
| <delay> | Optional:  The delay in milliseconds between each message publication. If you do not provide this option, the default is 1000ms (1s) | 2000 |

***Notes:***

* In you do provide any <infileX> option, the script sends a default text message in NON\_PERSISTENT mode. The payload of the message is “message<N>” where <N> varies from 1 to <count>.
  1. Sending Loop

If you provide a single “-infile <infile>” option and <infile> is the name of an existing folder, the script will:

* For i = 1 to <count>
  + For each file present in the folder which name ends with “.msg”, read the first message of the file and send it.

If you provide a single “-infile <infile>” option and <infile> is the name of an existing file, the script will:

* For i = 1 to <count>
  + Read the first message of the file and send it.

If you provide multiple “-infile <infile>” options all pointing to existing files, the script will:

* For i = 1 to <count>
  + Read the first message of the first file and send it. The next time this step is done, the next file is read and so on in a loop.

With this mode, you can send semi-random messages.

***Example:***

…/EMSQueueSender.sh … -infile 1.msg –infile 2.msg –infile 3.msg –count 4

*The script will send the message in this fashion:*

*1.msg*

*2.msg*

*3.msg*

*1.msg*

If you provide multiple “-infile <infile>” options pointing to existing files and folders, the script will fail when reading the folders.

1. EMSTestQueueConnection

EMSTools are a set of command line scripts for Windows and Unix. Each script usage is documented in its own chapter.

* 1. Description

**EMSTestQueueConnection** connects to the EMS server (with a queue connection) and then disconnects.

You can use **EMSTestQueueConnection** to test the connection parameters, connection factory names, credentials or SSL parameters.

* 1. Usage

From any working directory, type the following:

<install-folder>/EMSTools/bin/EMSTestQueueConnection.sh

<server-options>

<ssl-options>

<other-options>

Or the following command (on Windows):

<install-folder>\EMSTools\bin\EMSTestQueueConnection.bat

<server-options>

<ssl-options>

<other-options>

Options are listed in this table:

Table 9: EMSTestQueueConnection <options>

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Examples |
| <server-options> | Common Options to connect to the EMS Server (see chapter “3 Common Options” for more information). |  |
| <ssl-options> |  |
| <other-options> |  |

1. EMSTestTopicConnection

EMSTools are a set of command line scripts for Windows and Unix. Each script usage is documented in its own chapter.

* 1. Description

**EMSTestTopicConnection** connects to the EMS server (with a topic connection) and then disconnects.

You can use **EMSTestTopicConnection** to test the connection parameters, connection factory names, credentials or SSL parameters.

* 1. Usage

From any working directory, type the following command:

<install-folder>/EMSTools/bin/EMSTestTopicConnection.sh

<server-options>

<ssl-options>

<other-options>

Or the following command (on Windows):

<install-folder>\EMSTools\bin\EMSTestTopicConnection.bat

<server-options>

<ssl-options>

<other-options>

Options are listed in this table:

Table 10: EMSTestTopicConnection <options>

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Examples |
| <server-options> | Common Options to connect to the EMS Server (see chapter “3 Common Options” for more information). |  |
| <ssl-options> |  |
| <other-options> |  |

1. EMSAdmin

EMSTools are a set of command line scripts for Windows and Unix. Each script usage is documented in its own chapter.

* 1. Description

**EMSAdmin** connects to one TIBCO EMS server with the TIBCO EMS API and prints out on standard output some of the server details in CSV format. This can be redirected into a .csv file for further reading into Microsoft Excel for example.

The “-destroyConnection” command line option allows deleting a client connection on the EMS server for administration purposes.

* 1. Usage

From any working directory, type the following:

<install-folder>/EMSTools/bin/EMSAdmin.sh

Or the following command (on Windows):

<install-folder>\EMSTools\bin\EMSAdmin.bat

This will prints out all available command line options.

The connection options to the EMS server are not as extended as for the other tools. They are limited to the following:

-server <url>

-user <user>

-password <password>

Options are listed in this table:

Table 11: EMSAdmin <server-options>

| Parameter | Description | Examples |
| --- | --- | --- |
| <url> | The URL of the server. | tcp://localhost:2222  tcp://primary:1234 |
| <user> | The username/login to the server.  This user must have admin privileges on the server. | esb |
| <password> | The password to the server. | esb |

1. EMSAdmin-all

EMSTools are a set of command line scripts for Windows and Unix. Each script usage is documented in its own chapter.

* 1. Description

**EMSAdmin-all** connects to one TIBCO EMS server with the TIBCO EMS API and prints out all details of the server into a set of .csv files.

* 1. Usage

From any working directory, type the following command:

<install-folder>/EMSTools/bin/EMSAdmin-all.sh –server <url> -user <user> -password <password>

This script is not available on Windows.

The script will create a new folder under EMSTools folder to store all collected .csv files.

Options are listed in this table:

Table 12: EMSAdmin-all <server-options>

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Examples |
| <url> | The URL of the server. | tcp://localhost:2222  tcp://primary:1234 |
| <user> | The username/login to the server.  This user must have admin privileges on the server. | esb |
| <password> | The password to the server. | esb |

1. Frequently Asked Questions
   1. How to specify TIBCO JNDI reconnection parameters?

Add the following options into the command line:

-jndi\_property com.tibco.tibjms.connect.attemptcount=<X>

-jndi\_property com.tibco.tibjms.connect.attemptdelay=<Y>

-jndi\_property com.tibco.tibjms.connect.attempttimeout = <Z>

-jndi\_property com.tibco.tibjms.reconnect.attemptcount = <RX>

-jndi\_property com.tibco.tibjms.reconnect.attemptdelay=<RY>

-jndi\_property com.tibco.tibjms.reconnect.attempttimeout = <RZ>

**This option (-jndi\_property) is supported since version 1.3.3.**

* 1. How to connect to a Solace Appliance?

In the command line, you can specify your <server-options> as following:

-jndi\_provider\_factory com.solacesystems.jndi.SolJNDIInitialContextFactory

-jndi\_url TBC

-username <username>

-password <password>

If you need to specify a Solace VPN, add this:

-jndi\_property Solace\_JMS\_VPN=<VPN>

**This option (-jndi\_property) is supported since version 1.3.3.**

***Notes:***

• The **EMSAdmin** and **EMSAdmin-all** scripts will not work against a Solace Appliance as they use the proprietary TIBCO EMS Administration API.