



Ultra Messaging (Version 6.15)

C Examples

Contents

1	Introduction	5
1.1	C Examples Introduction	5
1.2	Configuring C Examples	6
1.3	Building C Examples	6
1.4	Unhandled C Events	6
1.5	C Example Support Files	7
1.6	Persistence Tutorial C Files	7
1.7	C Examples	8
1.7.1	Example lbmhfrcv.c	8
1.7.2	Example lbmhfrcvq.c	8
1.7.3	Example lbmhsrc.c	9
1.7.4	Example lbmhsrcv.c	9
1.7.5	Example lbmhtcv.c	11
1.7.6	Example lbmimsg.c	11
1.7.7	Example lbmireq.c	11
1.7.8	Example lbmlatping.c	12
1.7.9	Example lbmlatpong.c	12
1.7.10	Example lbmmon.c	12
1.7.11	Example lbmmoncache.c	14
1.7.12	Example lbmmon_cmd.c	15
1.7.13	Example lbmmondata.c	15
1.7.14	Example lbmmonudp.c	16
1.7.15	Example lbmmrcv.c	17
1.7.16	Example lbmmrcvq.c	18
1.7.17	Example lbmmreq.c	19
1.7.18	Example lbmmsrc.c	20
1.7.19	Example lbmpong.c	21
1.7.20	Example lbmprice.c	22
1.7.21	Example lbmrcv.c	22
1.7.22	Example lbmrcvq.c	24
1.7.23	Example lbmrcvxsp.c	25

1.7.24	Example lbmreq.c	25
1.7.25	Example lbmresp.c	26
1.7.26	Example lbmresping.c	26
1.7.27	Example lbmrespq.c	26
1.7.28	Example lbmspike.c	27
1.7.29	Example lbmsrc.c	27
1.7.30	Example lbmssrc.c	29
1.7.31	Example lbmssrcreq.c	30
1.7.32	Example lbmstrm.c	31
1.7.33	Example lbmtrreq.c	31
1.7.34	Example lbmwrcv.c	32
1.7.35	Example lbmwrcvq.c	33
1.7.36	Example minrcv.c	34
1.7.37	Example minrcv.cpp	34
1.7.38	Example minsrc.c	34
1.7.39	Example srs_cmd.c	34
1.7.40	Example srs_monitor_info_receiver.c	35
1.7.41	Example srs_monitor_info_receiver_json.c	35
1.7.42	Example tnwgdcmd.c	36
1.7.43	Example tnwgdmon.c	36
1.7.44	Example umedcmd.c	36
1.7.45	Example umedmon.c	37
1.7.46	Example ume-example-rcv-2.c	37
1.7.47	Example ume-example-rcv-3.c	37
1.7.48	Example ume-example-rcv.c	37
1.7.49	Example ume-example-src-2.c	37
1.7.50	Example ume-example-src-3.c	38
1.7.51	Example ume-example-src.c	38
1.7.52	Example umercv.c	38
1.7.53	Example umesnaprepo.c	38
1.7.54	Example umesrc.c	39
1.7.55	Example umessrc.c	39
1.7.56	Example umestored_example.c	41
1.7.57	Example umqrcv.c	41
1.7.58	Example umqsrc.c	42
1.8	Example Protocol Files	43
1.8.1	Example dro_mon.proto	43
1.8.2	Example srs_mon.proto	43
1.8.3	Example um_mon_attributes.proto	43
1.8.4	Example um_mon_control.proto	43

1.8.5	Example ump_mon.proto	43
1.8.6	Example ums_mon.proto	44

Chapter 1

Introduction

This document lists and gives some background information on the C-language example UM programs.

For policies and procedures related to Ultra Messaging Technical Support, see [UM Support](#).

(C) Copyright 2004,2022 Informatica LLC. All Rights Reserved.

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

This software is protected by patents as detailed at <https://www.informatica.com/legal/patents.html>.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, please report them to us in writing at Informatica LLC 2100 Seaport Blvd. Redwood City, CA 94063.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided.

INFORMATICA LLC PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

See **UM Glossary** for Ultra Messaging terminology, abbreviations, and acronyms.

1.1 C Examples Introduction

These programs were written to help in troubleshooting, testing, and demonstrating UM coding techniques. See also [Java Example Source Code](#) and [C# Example Source Code](#).

Since the tools are written to be useful as well as instructive, they are more complex than purely-instructive examples would be, with many options to add or subtract functionality. See [UMExamples](#) for purely-instructive examples of a variety of UM use cases.

The example C programs listed here are provided in both source form and in binary executable form.

1.2 Configuring C Examples

The example programs universally provide the "-c filename" command-line option. Using that option, the example application calls the **lbm_config()** API. However, note that this API is not recommended for use with XML-format LBM configuration files, largely because you are not able to specify an application name.

To use an XML configuration file with a UM example application, set the environment variables:

- LBM_XML_CONFIG_APPNAME - Desired name of application.
- LBM_XML_CONFIG_FILENAME - Path name of XML configuration file.

In this way, UM will correctly set the example application's name and will properly load the XML configuration file.

1.3 Building C Examples

Most users are not interested in compiling these sources in their current form, but instead use them for "spare parts", extracting fragments of code as appropriate. For users who wish to build the tools, and especially for users who may want to modify the tools, we recommend creating a new directory and copying all of the doc/example directory contents into that new directory.

The "Makefile.unix" is designed to be used by making a copy of the example directory under your binary package. For example, after installing the UMP 6.13 binary and documentation packages on your Linux system, you should have two directories named "UMP_6.13" and "UM_6.13_doc". To copy and build:

```
$ mkdir UMP_6.13/doc
$ cp -r UM_6.13_doc/doc/example UMP_6.13/doc/example
$ cd UMP_6.13/doc/example
$ make -f Makefile.unix TARGET_PLATFORM=Linux-glibc-2.17-x86_64 PRODUCT=UMP
```

Note

For UM versions prior to 6.14, the above command might fail on versions of Linux that do not have the "libnsl.so" library pre-installed. For example, CentOS 8. An easy workaround is to edit "Makefile.unix" and remove "-lnsl" from the LINUX_LIBS definition.

The Windows "nmake" file "Makefile.windows" is based on older versions of Visual Studio and may not be straightforward to make work as-is. It is still included for reference purposes. For example:

```
LBMSRC_OBJS = lbmsrc.obj verifymsg.obj getopt.obj
lbmsrc.exe:
```

indicates that the "lbmsrc.exe" program requires 3 source files: "lbmsrc.c", "verifymsg.c", and "getopt.c".

1.4 Unhandled C Events

Each of the example programs is written to demonstrate a subset of UM's total available functionality. For example, some programs are written to demonstrate **Streaming** functionality (e.g. lbmsrc), while other programs are written to demonstrate **Persistence** functionality (e.g. umesrc), while still other programs are written to demonstrate **Queuing** functionality (e.g. umqsrc).

UM is generally designed to be event-driven, with events being delivered to the programs through standard callbacks, like source callbacks and receiver callbacks. There are many events which are common across all streaming, persistence, and queuing. Other events are specific to persistence, and still other events are specific to queuing.

This can lead to example programs reporting "unknown" or "unhandled" events. For example, if the "lbmsrc" streaming program is run with a configuration file that enables persistence, UM will deliver events that are specific to persistence to the "lbmsrc" program. But "lbmsrc" is designed for streaming, and does not include code cases for persistence or queuing events. Maybe you should change your configuration to disable persistence, or you should be using the "umesrc" example program.

Similarly, the "umqsrc" program expects queuing functionality, and can report unhandled events if persistence is configured. Or "umesrc" can report unhandled events if queuing is configured.

If you see an unhandled event, it is generally reported as a number. You can see which event this corresponds to by looking up the number in:

- **C Receiver Events** for subscribing programs and
- **C Source Events** for publishing programs.

Once you understand the nature of the unhandled event, you can decide how to change your configuration or choose a different program.

1.5 C Example Support Files

There are several source files in the example directory that contain useful functions to the main example programs.

getopt.c - utility functions to parse command-line options (for Windows).

verifymsg.c - utility function to help some programs create verifiable messages.

monmodopts.h - common include file used by many of the example programs. It includes option information for monitoring functionality.

replgetopt.h - common include file used by many of the example programs. It includes definitions for alternate getopt functions.

verifymsg.h - common include file used by many of the example programs. It includes definitions for "verifymsg.c" (which needs to be linked into many programs).

srs_monitor_info_msg.c - Module used by the srs_monitor_info_receiver program.

srs_monitor_info_msg.h - Definitions for srs_monitor_info_msg.c module.

srs_cmd_msg.c - Module used by the srs_cmd program.

srs_cmd_msg.h - Definitions for srs_cmd_msg.c module.

Makefile.unix - see contents for instructions of setting up environment variables.

Makefile.windows - see contents for instructions of setting up make variables.

lbmondiag.pl - Reads UDP packets and process statistics. See [lbmmonudp.c](#) and [lbmondiag.pl](#) in the UM Operations Guide.

1.6 Persistence Tutorial C Files

See **Demonstrating Persistence** for information on these files.

ume-example-src.c - Initial source application used in the tutorial.

ume-example-rcv.c - Initial receiver application used in the tutorial.

ume-example-src-2.c - Source application modified to use a UMP persistent store.

ume-example-rcv-2.c - Receiver application modified to use a UMP persistent store.

ume-example-src-3.c - Modified source application used to demonstrate persistence.

ume-example-rcv-3.c - Modified receiver application used to demonstrate persistence.

ume-example-config.xml - Elementary persistent store configuration file used for the tutorial.

1.7 C Examples

1.7.1 Example lbmhfrvc.c

Source code: [lbmhfrvc.c](#)

Purpose: application that receives messages from a given topic using a single hot-failover receiver.

Usage: `lbmhfrvc [-AEhsvV] [-c filename] [-r msgs] [-U losslev] topic`

- A = display messages as ASCII text
- c filename = Use LBM configuration file filename.
Multiple config files are allowed.
Example: `'-c file1.cfg -c file2.cfg'`
- d, --msec-delay=NUM Implements a number of milliseconds sleep per message received
- E = exit after source ends
- h = help
- r msgs = delete receiver after msgs messages
- s = print statistics along with bandwidth
- S = Exit after source ends, print throughput summary
- v = be verbose about incoming messages (-v -v = be even more verbose)
- V = verify message contents

1.7.2 Example lbmhfrvcq.c

Source code: [lbmhfrvcq.c](#)

Purpose: `lbmhfrvc.c`: application that receives messages from a given topic using a single hot-failover receiver and an event queue.

Usage: `lbmhfrvcq [-EhsvV] [-c filename] [-r msgs] [-U losslev] topic`

- c filename = Use LBM configuration file filename.
Multiple config files are allowed.
Example: `'-c file1.cfg -c file2.cfg'`
- E = exit after source ends
- h = help
- r msgs = delete receiver after msgs messages
- s = print statistics along with bandwidth
- S = Exit after source ends, print throughput summary
- U losslev = exit after losslev % unrecoverable loss

-v = be verbose about incoming messages (-v -v = be even more verbose)
 -V = verify message contents

1.7.3 Example lbmhfsrc.c

Source code: [lbmhfsrc.c](#)

Purpose: application that sends to a given topic using a single hot-failover source.

Usage: lbmhfsrc [options] topic

Available options:

- c filename = Use LBM configuration file filename.
Multiple config files are allowed.
Example: '-c file1.cfg -c file2.cfg'
- d delay = delay sending for delay seconds after source creation
- h = help
- i init = start at message init instead of 0
- l len = send messages of len bytes
- L linger = linger for linger seconds before closing context
- M msgs = send msgs number of messages
- N NUM = send on channel NUM
- P msec = pause after each send msec milliseconds
- R [UM]DATA/RETR = Set transport type to LBT-R[UM], set data rate limit to DATA bits per second, and set retransmit rate limit to RETR bits per second. For both limits, the optional k, m, and g suffixes may be used. For example, '-R 1m/500k' is the same as '-R 1000000/500000'
- s sec = print stats every sec seconds
- t filename = use filename contents as a recording of message sequence numbers
- V = construct verifiable messages
- x bits = Use 32 or 64 bits for hot-failover sequence numbers

1.7.4 Example lbmhfxrcv.c

Source code: [lbmhfxrcv.c](#)

Purpose: application that receives messages from a given topic using a single hot-failover receiver across contexts (HFX).

Usage: lbmhfxrcv [-aACdEfHqsSvV] [-I interface] [-c filename] [-r msgs] [-U losslev] topic

Available options:

- a, --arrival-order deliver messages in the order that they arrive.
- A, --ascii display messages as ASCII text (-A -A = newlines after each msg)
- c, --config=FILE Use LBM configuration file FILE.
Multiple config files are allowed.
Example: '-c file1.cfg -c file2.cfg'
- C, --context-stats fetch context rather than receiver stats
- d, --deliver-dups Enable duplicate delivery
- E, --exit exit when source stops sending
- h, --help display this help and exit
- I, --iface=CIDR create a context on the interface specified by CIDR.
Multiple interfaces are allowed.
Example: '-I 10.29.1.0/24 -I 10.29.2.0/24'
- q, --eventq use an LBM event queue

-r, --msgs=NUM	exit after NUM messages
-O, --orderchecks	Enable message order checking
-s, --stats=NUM	print LBM statistics every NUM seconds
--max-sources=NUM	allow up to NUM sources (for statistics gathering purposes)
-S, --stop	exit when source stops sending, and print throughput summary
-U, --losslev=NUM	exit after NUM% unrecoverable loss
-v, --verbose	be verbose about incoming messages (-v -v = be even more verbose)
-V, --verify	verify message contents

Monitoring options:

--monitor-rcv=NUM	monitor receiver every NUM seconds
--monitor-ctx=NUM	monitor context every NUM seconds
--monitor-transport=TRANS	use monitor transport module TRANS TRANS may be 'lbm', 'lbmsnmp', or 'udp', default is 'lbm'
--monitor-transport-opts=OPTS	use OPTS as transport module options
--monitor-format=FMT	use monitor format module FMT FMT may be 'csv' or 'pb'
--monitor-format-opts=OPTS	use OPTS as format module options
--monitor-appid=ID	use ID as application ID string

Transport and format options are passed as name=value pairs, separated by a semicolon.

The entire option string should be enclosed in double-quotes.

LBM transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC default is /29west/statistics
allow_debug=VAL	VAL may be 'off' or 'on' defaults to 'off'

LBMSNMP transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC default is /29west/statistics

UDP transport options:

address=IP	send statistics to address IP
port=NUM	send to UDP port NUM default is 2933
mcgroup=GRP	send on multicast group GRP
bcaddress=IP	send statistics to broadcast address IP
ttl=NUM	send multicast statistics with TTL NUM default is 16

CSV format options:

separator=CHAR	separate CSV fields with character CHAR defaults to ',' Don't use a semicolon!
----------------	--

PB format options:

filters=FILE	use FILE that contains filter options
--------------	---------------------------------------

1.7.5 Example lbmhtrcv.c

Source code: [lbmhtrcv.c](#)

Purpose: application that receives from a collection of HyperTopic patterns.

Usage: lbmhtrcv [options] <patterns_file

Where 'patterns_file' is a simple text file, supplied as standard input, containing one pattern per line.

Available options:

-d msec	delete hypertopic receiver every msec milliseconds
-h, --help	display this help and exit
-p string	set hypertopic prefix to string
-q	use event queue
-s, --statistics	print statistics along with bandwidth
-v, --verbose	be verbose about incoming messages
-x	exit after all receivers deleted

1.7.6 Example lbmimsg.c

Source code: [lbmimsg.c](#)

Purpose: application that sends immediate messages (either unicast or multicast) as fast as possible, either to a topic, or send topicless.

Usage: lbmimsg [options] topic

lbmimsg [options] -o

Available options:

-c filename	= Use LBM configuration file filename. Multiple config files are allowed. Example: '-c file1.cfg -c file2.cfg'
-d delay	= delay sending for delay seconds after source creation
-h	= help
-l len	= send messages of len bytes
-L linger	= linger for linger seconds before closing context
-M msgs	= send msgs number of messages
-n num	= Append a number between 1 and num to topic
-o	= send topic-less immediate messages
-P msec	= pause after each send msec milliseconds
-R [UM]DATA/RETR	= Set transport type to LBT-R[UM], set data rate limit to DATA bits per second, and set retransmit rate limit to RETR bits per second. For both limits, the optional k, m, and g suffixes may be used. For example, '-R 1m/500k' is the same as '-R 1000000/500000'
-T target	= target for unicast immediate messages

1.7.7 Example lbmireq.c

Source code: [lbmireq.c](#)

Purpose: application that sends immediate message requests (multicast or unicast) to a given topic and waits for responses.

Usage: lbmireq [-hv] [-c filename] [-l len] [-L linger] [-P sec] [-r rate/pct] [-R requests] [-T target] [topic]

```

-h, --help                display this help and exit
-t, --transport=TRANS     use transport module TRANS
                           TRANS may be 'lbm', 'udp', or 'lbmsnmp', default is
                           'lbm'
--transport-opts=OPTS    use OPTS as transport module options
                           See the 'UM Operations Guide' section 'Monitoring
                           Transport Modules'.
-f, --format=FMT          use format module FMT
                           FMT may be 'csv' or 'pb'
--format-opts=OPTS       use OPTS as format module options
                           See the 'UM Operations Guide' section 'Monitoring
                           Format Modules'.

```

Transport and format options are passed as name=value pairs, separated by a semicolon.

The entire option string should be enclosed in double-quotes.

LBM transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

```

config=FILE              use LBM configuration file FILE
topic=TOPIC              receive statistics on topic TOPIC
                           default is /29west/statistics
wctopic=PATTERN          receive statistics on wildcard topic PATTERN
                           See https://communities.informatica.com/infakb/faq/5/Pages/80075.aspx
                           for guidelines on using wildcard topics. Also make sure
                           the statistics
                           topic namespace is disjoint from the data topic namespace.

```

LBMSNMP transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

```

config=FILE              use LBM configuration file FILE
topic=TOPIC              receive statistics on topic TOPIC
                           default is /29west/statistics
wctopic=PATTERN          receive statistics on wildcard topic PATTERN
                           See https://communities.informatica.com/infakb/faq/5/Pages/80075.aspx
                           for guidelines on using wildcard topics. Also make sure
                           the statistics
                           topic namespace is disjoint from the data topic namespace.

```

UDP transport options:

```

port=NUM                receive on UDP port NUM
                           default is 2933
interface=IP            receive multicast on interface IP
                           default is INADDR_ANY (0.0.0.0)
mcgroup=GRP             receive on multicast group GRP

```

CSV format options:

```

separator=CHAR          separate CSV fields with character CHAR
                           defaults to ','
                           Don't use a semicolon!
passthrough=VAL         VAL may be 'off', 'on' or 'convert'
                           defaults to 'off'

```

PB format options:

```

passthrough=VAL         VAL may be 'off', 'on' or 'convert'

```

defaults to 'off'

1.7.11 Example lbmmoncache.c

Source code: [lbmmoncache.c](#)

Purpose: example LBM statistics monitoring application.

Usage: lbmmoncache [options]

Available options:

-c, --config=FILE	Use LBM configuration file FILE. Multiple config files are allowed. Example: '-c file1.cfg -c file2.cfg'
-C, --cache-size=size	Set the cache size to 'size' entries
-h, --help	display this help and exit
-t, --transport=TRANS	use transport module TRANS TRANS may be 'lbm', 'udp', or 'lbmsnmp', default is 'lbm'
--transport-opts=OPTS	use OPTS as transport module options
-f, --format=FMT	use format module FMT FMT may be 'csv'
--format-opts=OPTS	use OPTS as format module options

Transport and format options are passed as name=value pairs, separated by a semicolon.

The entire option string should be enclosed in double-quotes.

LBM transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	receive statistics on topic TOPIC default is /29west/statistics
wctopic=PATTERN	receive statistics on wildcard topic PATTERN See https://communities.informatica.com/infakb/faq/5/Pages/80075.aspx for guidelines on using wildcard topics. Also make sure the statistics topic namespace is disjoint from the data topic namespace.

LBMSNMP transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	receive statistics on topic TOPIC default is /29west/statistics
wctopic=PATTERN	receive statistics on wildcard topic PATTERN See https://communities.informatica.com/infakb/faq/5/Pages/80075.aspx for guidelines on using wildcard topics. Also make sure the statistics topic namespace is disjoint from the data topic namespace.

UDP transport options:

port=NUM	receive on UDP port NUM default is 2933
interface=IP	receive multicast on interface IP


```

                                default is INADDR_ANY (0.0.0.0)
mcgroup=GRP                    receive on multicast group GRP

CSV format options:
separator=CHAR                separate CSV fields with character CHAR
                                defaults to ','
                                Don't use a semicolon!
passthrough=VAL               VAL may be 'off', 'on' or 'convert'
                                defaults to 'off'

PB format options:
passthrough=VAL               VAL may be 'off', 'on' or 'convert'
                                defaults to 'off'

```

1.7.12 Example lbmmon_cmd.c

Source code: [lbmmon_cmd.c](#)

Purpose: send unicast immediate control requests to an LBMMON publisher.

Usage: lbmmon_cmd -T target_string -C command [options]

Available options:

```

-c filename = Use LBM configuration file filename.
              Multiple config files are allowed.
              Example: '-c file1.cfg -c file2.cfg'
-C command  = command to send [required]
-d          = dump default filter options to stdout
-D data     = data for command (either -D or -F, not both)
              Example: '-C SET_INTERVAL -D 30'
-F filename = filename for command (either -D or -F, not both)
              Example: '-C SET_FILTER_OPTIONS -F filter.cfg'
-h          = help
-I id       = Application ID of node for command
              Example: '-C SNAP -N UMESTORE -I storeName'
-L linger   = linger for linger seconds before closing context
-N node     = node type for command
              Example: '-C SNAP -N CONTEXT'
-P sec      = pause for sec seconds after sending request to wait for response
-T target   = target string for unicast immediate requests [required]

```

1.7.13 Example lbmmondata.c

Source code: [lbmmondata.c](#)

Purpose: example LBM statistics monitoring application.

Usage: lbmmondata [-c filename] [-t topicname]

```

-c filename = Use LBM configuration file filename.
              Multiple config files are allowed.
              Example: '-c file1.cfg -c file2.cfg'
-t topicname = use topic topicname to receive statistics

```

1.7.14 Example lbmmonudp.c

Source code: [lbmmonudp.c](#)

Purpose: application that receives LBM statistics and forwards as CSV over UDP.

Usage: lbmmonudp [options]

Available options:

```
-3, --force-32bit      force all data values to fit within 32 bits
                        default is to use native data size
                        applies only to 64-bit platforms
-a, --address=IP       send CSV data to unicast address IP
-b, --broadcast=IP     send CSV data to broadcast address IP
-f, --format=FMT       use monitor format module FMT
                        FMT may be 'csv'
      --format-opts=OPTS use OPTS as format module options
-h, --help             display this help and exit
-i, --interface=IP     send multicast via interface IP
-m, --multicast=GRP    send CSV data to multicast group GRP
-p, --port=NUM         send CSV data on UDP port NUM
                        default is port 1234
-t, --transport=TRANS  use monitor transport module TRANS
                        TRANS may be 'lbm' or 'udp', default is 'lbm'
      --transport-opts=OPTS use OPTS as transport module options
-T, --ttl=NUM          send multicast with TTL NUM
                        default is 1
```

Transport and format options are passed as name=value pairs, separated by a semicolon.

The entire option string should be enclosed in double-quotes.

LBM transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

```
config=FILE           use LBM configuration file FILE
topic=TOPIC           receive statistics on topic TOPIC
                        default is /29west/statistics
wctopic=PATTERN       receive statistics on wildcard topic PATTERN
                        See https://communities.informatica.com/infakb/faq/5/Pages/80075.aspx
                        for guidelines on using wildcard topics. Also make sure
                        the statistics
                        topic namespace is disjoint from the data topic namespace.
```

LBMSNMP transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

```
config=FILE           use LBM configuration file FILE
topic=TOPIC           receive statistics on topic TOPIC
                        default is /29west/statistics
wctopic=PATTERN       receive statistics on wildcard topic PATTERN
                        See https://communities.informatica.com/infakb/faq/5/Pages/80075.aspx
                        for guidelines on using wildcard topics. Also make sure
                        the statistics
                        topic namespace is disjoint from the data topic namespace.
```

UDP transport options:

```
port=NUM              receive on UDP port NUM
                        default is 2933
interface=IP          receive multicast on interface IP
```

```

                                default is INADDR_ANY (0.0.0.0)
mcgroup=GRP                    receive on multicast group GRP

CSV format options:
separator=CHAR                separate CSV fields with character CHAR
                                defaults to ','
                                Don't use a semicolon!
passthrough=VAL              VAL may be 'off', 'on' or 'convert'
                                defaults to 'off'

PB format options:
passthrough=VAL              VAL may be 'off', 'on' or 'convert'
                                defaults to 'off'

```

1.7.15 Example lbmmrcv.c

Source code: [lbmmrcv.c](#)

Purpose: application that receives messages from a set of one or more topics.

Usage: lbmmrcv [options]

```

-B, --bufsize=#                Set receive socket buffer size to # (in MB)
-c, --config=FILE              Use LBM configuration file FILE.
                                Multiple config files are allowed.
                                Example: '-c file1.cfg -c file2.cfg'

-C, --contexts=NUM            use NUM lbm_context_t objects
-E, --exit                    exit and end upon receiving End-of-Stream notification
-e, --end-flag=FILE            clean up and exit when file FILE is created
-h, --help                    display this help and exit
-i, --initial-topic=NUM        use NUM as initial topic number
-o, --regid-offset=offset      use offset to calculate Registration ID
                                (as source registration ID + offset)
                                offset of 0 forces creation of regid by store

-L, --linger=NUM              linger for NUM seconds after done
-r, --root=STRING              use topic names with root of STRING
-R, --receivers=NUM            create NUM receivers
-s, --statistics               print statistics along with bandwidth
-v, --verbose                  be verbose
-V, --verify                   verify message contents

```

Monitoring options:

```

--monitor-rcv=NUM              monitor receiver every NUM seconds
--monitor-ctx=NUM              monitor context every NUM seconds
--monitor-transport=TRANS      use monitor transport module TRANS
                                TRANS may be 'lbm', 'lbmsnmp', or 'udp', default
                                is 'lbm'

--monitor-transport-opts=OPTS  use OPTS as transport module options
--monitor-format=FMT           use monitor format module FMT
                                FMT may be 'csv' or 'pb'

--monitor-format-opts=OPTS     use OPTS as format module options
--monitor-appid=ID             use ID as application ID string

```

Transport and format options are passed as name=value pairs, separated by a semicolon.

The entire option string should be enclosed in double-quotes.

LBM transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC
	default is /29west/statistics
allow_debug=VAL	VAL may be 'off' or 'on'
	defaults to 'off'

LBMSNMP transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where
 <scope> is one of context, source, receiver, wildcard_receiver, or event_queue
 <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC
	default is /29west/statistics

UDP transport options:

address=IP	send statistics to address IP
port=NUM	send to UDP port NUM
	default is 2933
mgroup=GRP	send on multicast group GRP
bcaddress=IP	send statistics to broadcast address IP
ttl=NUM	send multicast statistics with TTL NUM
	default is 16

CSV format options:

separator=CHAR	separate CSV fields with character CHAR
	defaults to ','
	Don't use a semicolon!

PB format options:

filters=FILE	use FILE that contains filter options
--------------	---------------------------------------

1.7.16 Example lbmmrcvq.c

Source code: [lbmmrcvq.c](#)

Purpose: application that receives messages from a set of one or more topics using event queues.

Usage: lbmmrcvq [options]

-B, --bufsize=#	Set receive socket buffer size to # (in MB)
-C, --config=FILE	Use LBM configuration file FILE.
	Multiple config files are allowed.
	Example: '-c file1.cfg -c file2.cfg'
-C, --contexts=NUM	use NUM lbm_context_t objects
-h, --help	display this help and exit
-i, --initial-topic=NUM	use NUM as initial topic number
-L, --linger=NUM	linger for NUM seconds after done
-r, --root=STRING	use topic names with root of STRING
-R, --receivers=NUM	create NUM receivers
-s, --statistics	print statistics along with bandwidth
-v, --verbose	be verbose

Monitoring options:

--monitor-rcv=NUM	monitor receiver every NUM seconds
--monitor-ctx=NUM	monitor context every NUM seconds
--monitor-transport=TRANS	use monitor transport module TRANS
	TRANS may be 'lbm', 'lbmsnmp', or 'udp', default is 'lbm'
--monitor-transport-opts=OPTS	use OPTS as transport module options
--monitor-format=FMT	use monitor format module FMT

	FMT may be 'csv' or 'pb'
--monitor-format-opts=OPTS	use OPTS as format module options
--monitor-appid=ID	use ID as application ID string
--monitor-evq=NUM	monitor event queue every NUM seconds

Transport and format options are passed as name=value pairs, separated by a semicolon.

The entire option string should be enclosed in double-quotes.

LBM transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC
	default is /29west/statistics
allow_debug=VAL	VAL may be 'off' or 'on'
	defaults to 'off'

LBMSNMP transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC
	default is /29west/statistics

UDP transport options:

address=IP	send statistics to address IP
port=NUM	send to UDP port NUM
	default is 2933
mcgroup=GRP	send on multicast group GRP
bcaddress=IP	send statistics to broadcast address IP
ttl=NUM	send multicast statistics with TTL NUM
	default is 16

CSV format options:

separator=CHAR	separate CSV fields with character CHAR
	defaults to ','
	Don't use a semicolon!

PB format options:

filters=FILE	use FILE that contains filter options
--------------	---------------------------------------

1.7.17 Example lbmmreq.c

Source code: [lbmmreq.c](#)

Purpose: application that sends request messages to a single topic and processes responses.

Usage: lbmmreq [options] topic

Available options:

-c filename	= Use LBM configuration file filename.
	Multiple config files are allowed.
	Example: '-c file1.cfg -c file2.cfg topicname'
-d delay	= delay sending for delay seconds after source creation
-h	= help
-l len	= send messages of len bytes

```
-r rate/pct = send with LBT-RM at rate and retransmission pct%
-R requests = send requests number of requests
-v = be verbose (-v -v = be even more verbose)
```

1.7.18 Example lbmmsrc.c

Source code: [lbmmsrc.c](#)

Purpose: send messages on multiple topics, optionally by multiple threads.

Topic names generated as a root, a dot, and by an integer.

By default, the first topic created will be '29west.example.multi.0'

Usage: lbmmsrc [options]

Available options:

-b, --batch=NUM	send messages in batch sizes of NUM between each pause
-c, --config=FILE	Use LBM configuration file FILE. Multiple config files are allowed. Example: '-c file1.cfg -c file2.cfg'
-d, --delay=NUM	delay sending for delay seconds after source creation
-h, --help	display this help and exit
-i, --initial-topic=NUM	use NUM as initial topic number [0]
-j, --late-join=NUM	enable Late Join with specified retention buffer size (in bytes)
-l, --length=NUM	send messages of length NUM bytes
-L, --linger=NUM	linger for NUM seconds after done
-M, --messages=NUM	send maximum of NUM messages
-P, --pause=NUM	pause NUM milliseconds after each send
-r, --root=STRING	use topic names with root of STRING [29west.example.multi]
-R, --rate=[UM]DATA/RETR	Set transport type to LBT-R[UM], set data rate limit to DATA bits per second, and set retransmit rate limit to RETR bits per second. For both limits, the optional k, m, and g suffixes may be used. For example, '-R 1m/500k' is the same as '-R 1000000/500000'
-s, --statistics=NUM	print stats every NUM seconds
-S, --sources=NUM	use NUM sources
-T, --threads=NUM	use NUM threads
-v, --verbose	be verbose
-V, --verifiable_msg	construct verifiable messages

Monitoring options:

--monitor-src=NUM	monitor source every NUM seconds
--monitor-ctx=NUM	monitor context every NUM seconds
--monitor-transport=TRANS	use monitor transport module TRANS TRANS may be 'lbm', 'lbmsnmp', or 'udp', default is 'lbm'
--monitor-transport-opts=OPTS	use OPTS as transport module options
--monitor-format=FMT	use monitor format module FMT FMT may be 'csv' or 'pb'
--monitor-format-opts=OPTS	use OPTS as format module options
--monitor-appid=ID	use ID as application ID string

Transport and format options are passed as name=value pairs, separated by a semicolon.

The entire option string should be enclosed in double-quotes.

LBM transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where
<scope> is one of context, source, receiver, wildcard_receiver, or event_queue
<option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

```

config=FILE          use LBM configuration file FILE
topic=TOPIC          send statistics on topic TOPIC
                     default is /29west/statistics
allow_debug=VAL      VAL may be 'off' or 'on'
                     defaults to 'off'

```

LBMSNMP transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

```

config=FILE          use LBM configuration file FILE
topic=TOPIC          send statistics on topic TOPIC
                     default is /29west/statistics

```

UDP transport options:

```

address=IP           send statistics to address IP
port=NUM             send to UDP port NUM
                     default is 2933
mcgroup=GRP          send on multicast group GRP
bcaddress=IP         send statistics to broadcast address IP
ttl=NUM              send multicast statistics with TTL NUM
                     default is 16

```

CSV format options:

```

separator=CHAR       separate CSV fields with character CHAR
                     defaults to ','
                     Don't use a semicolon!

```

PB format options:

```

filters=FILE          use FILE that contains filter options

```

1.7.19 Example lbmping.c

Source code: [lbmping.c](#)

Purpose: application that measures round trip message latency.

Usage: lbmping [-ChIqRv] [-c filename] [-i msgs] [-l len] [-M msgs] [-P msec] [-r rate/pct] [-s seed] [-t secs] [-T topic] id

-c filename = Use LBM configuration file filename.

Multiple config files are allowed.

Example: '-c file1.cfg -c file2.cfg'

-C = collect RTT data

-h = help

-i msgs = send and ignore msgs messages to warm up

-o offset = use offset to calculate Registration ID
(as source registration ID + offset)

offset of 0 forces creation of regid by store

-I = Use MIM

-l len = use len length messages

-M msgs = stop after receiving msgs messages

-P msec = pause after each send msec milliseconds

-q = use an LBM event queue

-r [UM]DATA/RETR = Set transport type to LBT-R[UM], set data rate limit to DATA bits per second, and set retransmit rate limit to RETR bits per second. For both limits, the optional k, m, and g suffixes may be used. For example, '-r 1m/500k' is the same as '-r 1000000/500000'

-R = perform RTT measurement per message

-s seed = init randomization of contents of message payload

```

-t secs = run for secs seconds
-T topic = topic name prefix (appended with '/' and id) [lbmpong]
-v = be verbose about each message
id = either 'ping' or 'pong'

```

1.7.20 Example lbmprice.c

Source code: [lbmprice.c](#)

Purpose: simulated price source and receiver for demonstration.

Usage: lbmprice -s [-h] [-c filename]

```

-c filename = Use LBM configuration file filename.
               Multiple config files are allowed.
               Example: '-c file1.cfg -c file2.cfg'

-h = help
-H = act has Hot Failover relay for a price source
-l pct = induce random receiver loss of pct percent
-n ms = set receiver NAK generation interval to ms milliseconds
-s = act as a price source (acts as a receiver by default)
-t ttl = set resolver (and multicast source) ttl to ttl
-v = be verbose

```

Alternate usage: lbmprice [-h] [-c filename]

```

-c filename = read config file
-h = help
-H = use Hot Failover receiver
-l pct = induce random receiver loss of pct percent, print max latency
-n ms = set receiver NAK generation interval to ms milliseconds
-o mode = set ordered delivery mode (1=ordered, 0=arrival order)
-t ttl = set resolver (and multicast source) ttl to ttl
-v = be verbose

```

1.7.21 Example lbmrcv.c

Source code: [lbmrcv.c](#)

Purpose: application that receives messages from a given topic.

Usage: lbmrcv [-ACEfhqsSvV] [-c filename] [-r msgs] [-U losslev] topic

Available options:

-A, --ascii	display messages as ASCII text (-A -A = newlines after each msg)
-c, --config=FILE	Use LBM configuration file FILE. Multiple config files are allowed. Example: '-c file1.cfg -c file2.cfg'
-C, --context-stats	fetch context rather than receiver stats
-E, --exit	exit when source stops sending
-f, --failover	use a hot-failover receiver
-h, --help	display this help and exit
-q, --eventq	use an LBM event queue
-r, --msgs=NUM	exit after NUM messages
-O, --orderchecks	Enable message order checking
-N, --channel=NUM	subscribe to channel NUM
-s, --stats=NUM	print LBM statistics every NUM seconds
--max-sources=NUM	allow up to NUM sources (for statistics gathering purposes)
-S, --stop	exit when source stops sending, and print throughput summary

```

-U, --losslev=NUM    exit after NUM% unrecoverable loss
-v, --verbose        be verbose about incoming messages (-v -v = be even more
                    verbose)
-V, --verify         verify message contents

```

Monitoring options:

```

--monitor-rcv=NUM      monitor receiver every NUM seconds
--monitor-ctx=NUM      monitor context every NUM seconds
--monitor-transport=TRANS
                    use monitor transport module TRANS
                    TRANS may be 'lbm', 'lbmsnmp', or 'udp', default
                    is 'lbm'
--monitor-transport-opts=OPTS
                    use OPTS as transport module options
--monitor-format=FMT   use monitor format module FMT
                    FMT may be 'csv' or 'pb'
--monitor-format-opts=OPTS
                    use OPTS as format module options
--monitor-appid=ID     use ID as application ID string

```

Transport and format options are passed as name=value pairs, separated by a semicolon.

The entire option string should be enclosed in double-quotes.

LBM transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

```

config=FILE          use LBM configuration file FILE
topic=TOPIC          send statistics on topic TOPIC
                    default is /29west/statistics
allow_debug=VAL      VAL may be 'off' or 'on'
                    defaults to 'off'

```

LBMSNMP transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

```

config=FILE          use LBM configuration file FILE
topic=TOPIC          send statistics on topic TOPIC
                    default is /29west/statistics

```

UDP transport options:

```

address=IP           send statistics to address IP
port=NUM             send to UDP port NUM
                    default is 2933
mcgroup=GRP          send on multicast group GRP
bcaddress=IP         send statistics to broadcast address IP
ttl=NUM              send multicast statistics with TTL NUM
                    default is 16

```

CSV format options:

```

separator=CHAR       separate CSV fields with character CHAR
                    defaults to ','
                    Don't use a semicolon!

```

PB format options:

```

filters=FILE         use FILE that contains filter options

```

1.7.22 Example lbmrcvq.c

Source code: [lbmrcvq.c](#)

Purpose: application that receives messages from a given topic using an event queue.

Usage: lbmrcvq [options] topic

Available options:

-C, --config=FILE	Use LBM configuration file FILE.
-C, --context-stats	fetch context rather than receiver stats
	Multiple config files are allowed.
	Example: '-c file1.cfg -c file2.cfg'
-E, --exit	exit after source ends
-h, --help	display this help and exit
-r NUM	delete receiver after NUM messages
-s, --stats=NUM	print LBM statistics every NUM seconds
-S, --stop	exit after source ends, print throughput summary
-v, --verbose	be verbose about incoming messages (-v -v = be even more verbose)
-V, --verify	verify message contents

Monitoring options:

--monitor-rcv=NUM	monitor receiver every NUM seconds
--monitor-ctx=NUM	monitor context every NUM seconds
--monitor-transport=TRANS	use monitor transport module TRANS
	TRANS may be 'lbm', 'lbmsnmp', or 'udp', default is 'lbm'
--monitor-transport-opts=OPTS	use OPTS as transport module options
--monitor-format=FMT	use monitor format module FMT
	FMT may be 'csv' or 'pb'
--monitor-format-opts=OPTS	use OPTS as format module options
--monitor-appid=ID	use ID as application ID string
--monitor-evq=NUM	monitor event queue every NUM seconds

Transport and format options are passed as name=value pairs, separated by a semicolon.

The entire option string should be enclosed in double-quotes.

LBM transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC
	default is /29west/statistics
allow_debug=VAL	VAL may be 'off' or 'on'
	defaults to 'off'

LBMSNMP transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC
	default is /29west/statistics

UDP transport options:

address=IP	send statistics to address IP
port=NUM	send to UDP port NUM
	default is 2933
mcgroup=GRP	send on multicast group GRP
bcaddress=IP	send statistics to broadcast address IP

```

ttl=NUM                send multicast statistics with TTL NUM
                        default is 16

CSV format options:
separator=CHAR          separate CSV fields with character CHAR
                        defaults to ','
                        Don't use a semicolon!

PB format options:
filters=FILE            use FILE that contains filter options

```

1.7.23 Example lbmrcvxsp.c

Source code: [lbmrcvxsp.c](#)

Purpose: application that receives messages from a given topic, mapping transports to various XSPs.

Usage: lbmrcv [-ACdDEhPrRsSv] [-c filename] [-r msgs] topic

Available options:

```

-A, --ascii                display messages as ASCII text (-A -A = newlines
                           after each msg)
-c, --config=FILE          Use LBM configuration file FILE.
                           Multiple config files are allowed.
                           Example: '-c file1.cfg -c file2.cfg'
-C, --context-stats        fetch context rather than receiver stats
-d, --defer-xsp-deletion   don't delete xsp until shutdown
-D, --default-xsp          use the default XSP for all transports
-E, --exit                 exit when source stops sending
-h, --help                 display this help and exit
-P, --round-robin-preallocate preallocate the XSPs - use with -R
-Q, --sequential-xsp      use sequential mode for XSPs
-r, --msgs=NUM             exit after NUM messages
-R, --round-robin=NUM      use a simple round-robin method for assigning
                           transports to NUM XSPs.
                           (this is the DEFAULT for this application, with a
                           NUM of 3)
-s, --stats=NUM            print LBM statistics every NUM seconds
-S, --stop                 exit when source stops sending, and print
                           throughput summary
-v, --verbose              be verbose about incoming messages (-v -v = be
                           even more verbose)
-V, --verify               verify message contents

```

1.7.24 Example lbmreq.c

Source code: [lbmreq.c](#)

Purpose: application that sends requests on a single topic and waits for responses.

Usage: lbmreq [options] topic

Available options:

```

-c filename = Use LBM configuration file filename.
               Multiple config files are allowed.
               Example: '-c file1.cfg -c file2.cfg'
-d sec = delay sending for delay seconds after source creation

```

```

-h = help
-l len = send messages of len bytes
-L linger = linger for linger seconds before closing context
-P sec = pause sec seconds after sending request for responses to arrive
-r [UM]DATA/RETR = Set transport type to LBT-R[UM], set data rate limit to
                    DATA bits per second, and set retransmit rate limit to
                    RETR bits per second. For both limits, the optional
                    k, m, and g suffixes may be used. For example,
                    '-r 1m/500k' is the same as '-r 1000000/500000'
-R requests = send requests number of requests
-q = Use Event Queue
-v = be verbose (-v -v = be even more verbose)

```

1.7.25 Example lbmresp.c

Source code: [lbmresp.c](#)

Purpose: application that receives request messages on a single topic and sends responses back

```

Usage: lbmresp [-Ehsv] [-c filename] [-l len] [-r responses] [-f topic] topic
      -c filename = Use LBM configuration file filename.
                   Multiple config files are allowed.
                   Example: '-c file1.cfg -c file2.cfg'
      -E = end after end-of-stream
      -h = help
      -l len = use len bytes for the length of each response
      -r responses = send responses messages for each request
      -s = be silent about incoming messages
      -v = be verbose (-v -v = be even more verbose)
      -f = forward request to responders listening on given topic

```

1.7.26 Example lbmresping.c

Source code: [lbmresping.c](#)

Purpose: Application that tests the operation of lbm topic resolution by creating a source and reporting time required for it to resolve and join the source.

```

Usage: lbmresping [-h] [-c filename] [unicast_resolver_host]
      -c filename = Use LBM configuration file filename.
                   Multiple config files are allowed.
                   Example: '-c file1.cfg -c file2.cfg'
      -h = help

```

1.7.27 Example lbmrespq.c

Source code: [lbmrespq.c](#)

Purpose: application that receives request messages on a single topic and sends responses back, using an event queue.

```

Usage: lbmrespq [-hs] [-c filename] [-r msgs] topic

```

```

-c filename = Use LBM configuration file filename.
               Multiple config files are allowed.
               Example: '-c file1.cfg -c file2.cfg'
-h = help
-P msec = pause msec milliseconds before sending response
-r msgs = delete receiver after msgs request messages
-s = be silent about requests/sec rate
-v = be verbose (-v -v = be even more verbose)

```

1.7.28 Example lbmspike.c

Source code: [lbmspike.c](#)

Purpose: application that generates & receives message spikes for performance testing.

Usage: lbmspike -R [-dhq] [-c filename] [-o ord] [-u bufsiz] [topic]

```

-c filename = Use LBM configuration file filename.
               Multiple config files are allowed.
               Example: '-c file1.cfg -c file2.cfg'
-d = dump message time stamps to a file
-h = help
-o ord = set receiver ordered delivery to ord
-q = process received messages on an event queue
-R = role is receiver (default role is source)
-u bufsiz = UDP buffer size for LBT-RM

```

Alternate usage: lbmspike [-dhLn] [-B bghumms] [-c filename] [-l len] [-M msgs] [-r rate/pct] [-v recovms] [topic]

```

-B bghumms = milliseconds between "background hum" messages
-c filename = read config file filename
-d = dump message time stamps to a file
-h = help
-l len = use len length messages
-L = use TCP-LB
-M msgs = stop after receiving msgs messages
-n = use non-blocking writes
-r [UM]DATA/RETR = Set transport type to LBT-R[UM], set data rate limit to
                   DATA bits per second, and set retransmit rate limit to
                   RETR bits per second. For both limits, the optional
                   k, m, and g suffixes may be used. For example,
                   '-r 1m/500k' is the same as '-r 1000000/500000'
-v recovms = milliseconds after spike to allow for recovery

```

1.7.29 Example lbmsrc.c

Source code: [lbmsrc.c](#)

Purpose: application that sends to a single topic as fast as possible.

Usage: lbmsrc [options] topic

Available options:

```

-c, --config=FILE      Use LBM configuration file FILE.
                       Multiple config files are allowed.
                       Example: '-c file1.cfg -c file2.cfg'
-d, --delay=NUM        delay sending for NUM seconds after source creation
-h, --help              display this help and exit

```

-j, --late-join=NUM	enable Late Join with specified retention buffer size (in bytes)
-l, --length=NUM	send messages of NUM bytes
-L, --linger=NUM	linger for NUM seconds before closing context
-M, --messages=NUM	send NUM messages
-n, --non-block	use non-blocking I/O
-N, --channel=NUM	send on channel NUM
-P, --pause=NUM	pause NUM milliseconds after each send
-R, --rate=[UM]DATA/RETR	Set transport type to LBT-R[UM], set data rate limit to DATA bits per second, and set retransmit rate limit to RETR bits per second. For both limits, the optional k, m, and g suffixes may be used. For example, '-R 1m/500k' is the same as '-R 1000000/500000'
-s, --statistics=NUM	print statistics every NUM seconds
-v, --verbose	be verbose about each message
-V, --verifiable	construct verifiable messages

Monitoring options:

--monitor-src=NUM	monitor source every NUM seconds
--monitor-ctx=NUM	monitor context every NUM seconds
--monitor-transport=TRANS	use monitor transport module TRANS TRANS may be 'lbm', 'lbmsnmp', or 'udp', default is 'lbm'
--monitor-transport-opts=OPTS	use OPTS as transport module options
--monitor-format=FMT	use monitor format module FMT FMT may be 'csv' or 'pb'
--monitor-format-opts=OPTS	use OPTS as format module options
--monitor-appid=ID	use ID as application ID string

Transport and format options are passed as name=value pairs, separated by a semicolon.

The entire option string should be enclosed in double-quotes.

LBM transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC default is /29west/statistics
allow_debug=VAL	VAL may be 'off' or 'on' defaults to 'off'

LBMSNMP transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC default is /29west/statistics

UDP transport options:

address=IP	send statistics to address IP
port=NUM	send to UDP port NUM default is 2933
mcgroup=GRP	send on multicast group GRP
bcaddress=IP	send statistics to broadcast address IP
ttl=NUM	send multicast statistics with TTL NUM default is 16

CSV format options:

```
separator=CHAR          separate CSV fields with character CHAR
                        defaults to ','
                        Don't use a semicolon!
```

PB format options:

```
filters=FILE            use FILE that contains filter options
```

1.7.30 Example lbmssrc.c

Source code: [lbmssrc.c](#)

Purpose: application that uses Smart Source to send to a single topic.

Usage: lbmssrc [options] topic

Available options:

```
-a, --available-data-space  print the length of available data space
-b, --user-supplied-buffer  send messages using a user-supplied buffer
-c, --config=FILE          Use LBM configuration file FILE.
                           Multiple config files are allowed.
                           Example: '-c file1.cfg -c file2.cfg'
-d, --delay=NUM            delay sending for NUM seconds after source creation
-h, --help                 display this help and exit
-i, --int-mprop=VAL,KEY    send integer message property value VAL with name KEY
-j, --late-join=NUM        enable Late Join with specified retention buffer count
-l, --length=NUM           send messages of NUM bytes
                           NOTE: set LBM_SMART_SOURCE_CHECK=0xffffffff env
                           variable
                           to check if the maximum message length is
                           exceeded
-L, --linger=NUM           linger for NUM seconds before closing context
-M, --messages=NUM         send NUM messages
-N, --channel=NUM          send on channel NUM
-S, --perf-stats=NUM,OT    print performance stats every NUM messages sent
                           If optional OT is given, override the default 10 usec
                           Outlier Threshold
-P, --pause=NUM            pause NUM milliseconds after each send
-s, --statistics=NUM       print statistics every NUM seconds
-v, --verbose              be verbose; add per message data
-V, --verifiable           construct verifiable messages
```

Monitoring options:

```
--monitor-src=NUM          monitor source every NUM seconds
--monitor-ctx=NUM          monitor context every NUM seconds
--monitor-transport=TRANS  use monitor transport module TRANS
                           TRANS may be 'lbm', 'lbmsnmp', or 'udp', default
                           is 'lbm'
--monitor-transport-opts=OPTS use OPTS as transport module options
--monitor-format=FMT       use monitor format module FMT
                           FMT may be 'csv' or 'pb'
--monitor-format-opts=OPTS use OPTS as format module options
--monitor-appid=ID         use ID as application ID string
```

Transport and format options are passed as name=value pairs, separated by a semicolon.

The entire option string should be enclosed in double-quotes.

LBM transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC
	default is /29west/statistics
allow_debug=VAL	VAL may be 'off' or 'on'
	defaults to 'off'

LBMSNMP transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where
 <scope> is one of context, source, receiver, wildcard_receiver, or event_queue
 <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC
	default is /29west/statistics

UDP transport options:

address=IP	send statistics to address IP
port=NUM	send to UDP port NUM
	default is 2933
mgroup=GRP	send on multicast group GRP
bcaddress=IP	send statistics to broadcast address IP
ttl=NUM	send multicast statistics with TTL NUM
	default is 16

CSV format options:

separator=CHAR	separate CSV fields with character CHAR
	defaults to ','
	Don't use a semicolon!

PB format options:

filters=FILE	use FILE that contains filter options
--------------	---------------------------------------

1.7.31 Example lbmssrcreq.c

Source code: [lbmssrcreq.c](#)

Purpose: application that sends requests on a single topic and waits for responses.

Usage: lbmssrcreq [options] topic

Available options:

- a, --available-data-space print the length of available data space
- b, --user-supplied-buffer send messages using a user-supplied buffer
- c filename = Use LBM configuration file filename.
 Multiple config files are allowed.
 Example: '-c file1.cfg -c file2.cfg'
- d sec = delay sending for delay seconds after source creation
- h = help
- l len = send messages of len bytes
- L linger = linger for linger seconds before closing context
- P sec = pause sec seconds after sending request for responses to arrive
- r [UM]DATA/RETR = Set transport type to LBT-R[UM], set data rate limit to
 DATA bits per second, and set retransmit rate limit to
 RETR bits per second. For both limits, the optional
 k, m, and g suffixes may be used. For example,
 '-r 1m/500k' is the same as '-r 1000000/500000'
- R requests = send requests number of requests
- v = be verbose (-v -v = be even more verbose)

1.7.32 Example lbmstrm.c

Source code: [lbmstrm.c](#)

Purpose: application that sends messages to one or more topics at a specified rate.

Usage: lbmstrm [options]

Topic names generated as a root, followed by a dot, followed by an integer.

By default, the first topic created will be '29west.example.multi.0'

Available options:

-c, --config=FILE	Use LBM configuration file FILE. Multiple config files are allowed. Example: '-c file1.cfg -c file2.cfg'
-h, --help	display this help and exit
-H, --hf	Use hot failover sources
-i, --initial-topic=NUM	use NUM as initial topic number [0]
-j, --late-join=NUM	enable Late Join with specified retention buffer size (in bytes)
-l, --length=NUM	send messages of length NUM bytes [25]
-L, --linger=NUM	linger for NUM seconds after done [10]
-m, --message-rate=NUM	send at NUM messages per second [10000]
-M, --messages=NUM	send maximum of NUM messages [10000000]
-r, --root=STRING	use topic names with root of STRING [29west.example.multi]
-R, --rate=[UM]DATA/RETR	Set transport type to LBT-R[UM], set data rate limit to DATA bits per second, and set retransmit rate limit to RETR bits per second. For both limits, the optional k, m, and g suffixes may be used. For example, '-R 1m/500k' is the same as '-R 1000000/500000'
-s, --statistics=NUM	print stats every NUM seconds
-S, --sources=NUM	use NUM sources [100]
-t, --tight	tight loop (cpu-bound) for even message spacing
-T, --threads=NUM	use NUM threads [1]
-x, --bits=NUM	use NUM bits for hot failover sequence number size (32 or 64)

1.7.33 Example lbmtrreq.c

Source code: [lbmtrreq.c](#)

Purpose: application that invokes the Topic Resolution Request API.

Usage: lbmtrreq [options]

Available options:

-c, --config=FILE	Use LBM configuration file FILE. Multiple config files are allowed. Example: '-c file1.cfg -c file2.cfg'
-a, --adverts	Request Advertisements
-q, --queries	Request Queries
-w, --wildcard	Request Wildcard Queries
-A, --ctx-ads	Request Context Advertisements
-Q, --ctx-queries	Request Context Queries
-I, --gw-interest	Request Gateway Interest
-i, --interval=NUM	Interval between request
-d, --duration=NUM	Minimum duration of requests
-L, --linger=NUM	Linger for NUM seconds before closing context

1.7.34 Example lbmwrvc.c

Source code: [lbmwrvc.c](#)

Purpose: application that receives messages from a wildcard receiver.

Usage: lbmwrvc [options] pattern

Available options:

-c, --config=FILE	Use LBM configuration file FILE. Multiple config files are allowed. Example: '-c file1.cfg -c file2.cfg'
-D, --deregister	Send Deregistration after receiving 1000 messages
-E, --exit	exit after source ends
-h, --help	display this help and exit
-q	Use event queue
-r NUM	delete receiver after NUM messages
-N, --channel=NUM	subscribe to channel NUM
-s, --statistics	print statistics along with bandwidth
-v, --verbose	be verbose about incoming messages (-v -v = be even more verbose)

Monitoring options:

--monitor-ctx=NUM	monitor context every NUM seconds
--monitor-transport=TRANS	use monitor transport module TRANS TRANS may be 'lbm', 'lbmsnmp', or 'udp', default is 'lbm'
--monitor-transport-opts=OPTS	use OPTS as transport module options
--monitor-format=FMT	use monitor format module FMT FMT may be 'csv' or 'pb'
--monitor-format-opts=OPTS	use OPTS as format module options
--monitor-appid=ID	use ID as application ID string

Transport and format options are passed as name=value pairs, separated by a semicolon.

The entire option string should be enclosed in double-quotes.

LBM transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC default is /29west/statistics
allow_debug=VAL	VAL may be 'off' or 'on' defaults to 'off'

LBMSNMP transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC default is /29west/statistics

UDP transport options:

address=IP	send statistics to address IP
port=NUM	send to UDP port NUM default is 2933
mcgroup=GRP	send on multicast group GRP
bcaddress=IP	send statistics to broadcast address IP
ttl=NUM	send multicast statistics with TTL NUM default is 16

CSV format options:

```
separator=CHAR          separate CSV fields with character CHAR
                        defaults to ','
                        Don't use a semicolon!
```

PB format options:

```
filters=FILE            use FILE that contains filter options
```

1.7.35 Example lbmwrvcvq.c

Source code: [lbmwrvcvq.c](#)

Purpose: application that receives messages from a wildcard receiver, using an event queue.

Usage: lbmwrvcvq [options] pattern

Available options:

```
-C, --config=FILE      Use LBM configuration file FILE.
                        Multiple config files are allowed.
                        Example: '-c file1.cfg -c file2.cfg'

-E, --exit             exit after source ends

-h, --help            display this help and exit

-r NUM                delete receiver after NUM messages

-s, --statistics      print statistics along with bandwidth

-v, --verbose         be verbose about incoming messages (-v -v = be even more
                        verbose)
```

Monitoring options:

```
--monitor-ctx=NUM      monitor context every NUM seconds

--monitor-transport=TRANS
                        use monitor transport module TRANS
                        TRANS may be 'lbm', 'lbmsnmp', or 'udp', default
                        is 'lbm'

--monitor-transport-opts=OPTS
                        use OPTS as transport module options

--monitor-format=FMT   use monitor format module FMT
                        FMT may be 'csv' or 'pb'

--monitor-format-opts=OPTS
                        use OPTS as format module options

--monitor-appid=ID     use ID as application ID string

--monitor-evq=NUM      monitor event queue every NUM seconds
```

Transport and format options are passed as name=value pairs, separated by a semicolon.

The entire option string should be enclosed in double-quotes.

LBM transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

```
config=FILE           use LBM configuration file FILE
topic=TOPIC           send statistics on topic TOPIC
                        default is /29west/statistics
allow_debug=VAL       VAL may be 'off' or 'on'
                        defaults to 'off'
```

LBMSNMP transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

```
config=FILE           use LBM configuration file FILE
topic=TOPIC           send statistics on topic TOPIC
```

```

                                default is /29west/statistics

UDP transport options:
  address=IP                    send statistics to address IP
  port=NUM                      send to UDP port NUM
                                default is 2933
  mcgroup=GRP                  send on multicast group GRP
  bcaddress=IP                 send statistics to broadcast address IP
  ttl=NUM                      send multicast statistics with TTL NUM
                                default is 16

CSV format options:
  separator=CHAR               separate CSV fields with character CHAR
                                defaults to ','
                                Don't use a semicolon!

PB format options:
  filters=FILE                 use FILE that contains filter options

```

1.7.36 Example minrcv.c

Source code: [minrcv.c](#)

minrcv.c: minimal application that receives messages from a given topic.

1.7.37 Example minrcv.cpp

Source code: [minrcv.cpp](#)

minimal C++ application that receives messages from a given topic.

1.7.38 Example minsrc.c

Source code: [minsrc.c](#)

minsrc.c: minimal application that sends to a given topic.

1.7.39 Example srs_cmd.c

Source code: [srs_cmd.c](#)

Purpose: send unicast immediate command messages to an SRS daemon.

Usage: srs_cmd [options] [command_string]

Available options:

```

  -c filename = Use LBM configuration file filename.
                Multiple config files are allowed.
                Example: '-c file1.cfg -c file2.cfg'

```

```

-h = help
-L linger = linger for linger seconds before closing context
-T target = target for unicast immediate messages (required)
*****
*      help (print this message): h                               *
*      quit (exit application): q                                 *
*      report SRS version: version                               *
* set category publishing interval: srs_stats 0 | 200-N          *
*                               um_client_stats 0 | 200-N        *
*                               connection_events 0 | 200-N      *
*                               srs_error_stats 0 | 200-N        *
*                               um_client_error_stats 0 | 200-N  *
*                               config_opts 0 | 200-N            *
*                               internal_config_opts 0 | 200-N   *
* set all publishing intervals: interval 0 | 200-N              *
*      snapshot category: snap srs_stats | um_client_stats |   *
*                               connection_events | srs_error_stats |
*                               um_client_error_stats | config_opts |
*                               internal_config_opts             *
*      snapshot all categories: snap                             *
*****

```

1.7.40 Example srs_monitor_info_receiver.c

Source code: [srs_monitor_info_receiver.c](#)

Purpose: receive SRS monitor info messages on the specified topic.

Note: this tool is deprecated.

See <https://ultramessaging.github.io/currdoc/doc/ChangeLog/deprecations.html#deprecationsfor615>

Usage: srs_monitor_info_receiver [options] topic

Available options:

```

-c, --config=FILE      Use LBM configuration file filename.
                        Multiple config files are allowed.
                        Example: '-c file1.cfg -c file2.cfg'
-E, --exit              exit upon EOS reception
-h, --help              help
-L, --linger            linger for linger seconds before closing context

```

1.7.41 Example srs_monitor_info_receiver_json.c

Source code: [srs_monitor_info_receiver_json.c](#)

Purpose: receive and decode JSON SRS monitor info messages on the specified topic.

Note: this tool is not shipped in binary form and is deprecated.

See <https://ultramessaging.github.io/currdoc/doc/ChangeLog/deprecations.html#deprecationsfor615>

Usage: srs_monitor_info_receiver_json [options] topic

Available options:

```

-c, --config=FILE      Use LBM configuration file filename.
                        Multiple config files are allowed.
                        Example: '-c file1.cfg -c file2.cfg'
-E, --exit              exit upon EOS reception
-h, --help              help
-L, --linger            linger for linger seconds before closing context

```

1.7.42 Example tnwgdcmd.c

Source code: [tnwgdcmd.c](#)

Purpose: application sends unicast immediate command messages to a tnwgd publishing daemon.

Usage: `tnwgdcmd -T target_string -c config_file [command_string]`

Available options:

```
-c filename = Use LBM configuration file filename.
               Multiple config files are allowed.
               Example: '-c file1.cfg -c file2.cfg'

-h = help
-L linger = linger for linger seconds before closing context
-T target = target for unicast immediate messages (mandatory)
```

```
*****
* help (print this message): h                                     *
* quit (exit application): q                                       *
* set publishing interval: (0-N = interval in seconds)             *
*           ri 0-N          (routing info)                         *
*           gcfg 0-N        (gateway config)                       *
*           ["portal name"] pcfg 0-N    (portal config)            *
*           ["portal name"] pstat 0-N   (portal stats)             *
*           mallinfo 0-N    (malloc info)                          *
*                                                                     *
* snapshot all groups (and all portals) : snap                     *
* snapshot single group: snap (ri|gcfg|pcfg|pstat|mallinfo)        *
* snapshot single portal: "portal name" snap pcfg|pstat           *
* Print the current version of the monitor: version                *
*****
```

1.7.43 Example tnwgdmmon.c

Source code: [tnwgdmmon.c](#)

Purpose: application that receives DRO daemon messages on the specified publishing topic.

Usage: `tnwgdmmon [-Ehv] [-c filename] publishing_topic`

Available options:

```
-c, --config=FILE      Use LBM configuration file FILE.
                        Multiple config files are allowed.
                        Example: '-c file1.cfg -c file2.cfg'

-E, --exit             exit when source stops sending
-h, --help             display this help and exit
-v, --verbose          be verbose about incoming messages (-v -v = be even more
                        verbose)
```

1.7.44 Example umedcmd.c

Source code: [umedcmd.c](#) See **umedcmd Man Page** for usage information.

1.7.45 Example umedmon.c

Source code: [umedmon.c](#)

Purpose: application that receives umestore daemon messages on the specified publishing topic.

Usage: umedmon [-Ehv] [-c filename] publishing_topic

Available options:

-c, --config=FILE	Use LBM configuration file FILE. Multiple config files are allowed. Example: '-c file1.cfg -c file2.cfg'
-E, --exit	exit when source stops sending
-h, --help	display this help and exit
-v, --verbose	be verbose about incoming messages (-v -v = be even more verbose)

1.7.46 Example ume-example-rcv-2.c

Source code: [ume-example-rcv-2.c](#)

ume-example-rcv-2.c: - Persistent example receiver program.
See Persistence Guide document.

1.7.47 Example ume-example-rcv-3.c

Source code: [ume-example-rcv-3.c](#)

ume-example-rcv-3.c: - Persistent example receiver program.
See Persistence Guide document.

1.7.48 Example ume-example-rcv.c

Source code: [ume-example-rcv.c](#)

ume-example-rcv.c: - Persistent example receiver program.
See Persistence Guide document.

1.7.49 Example ume-example-src-2.c

Source code: [ume-example-src-2.c](#)

ume-example-src-2.c: - Persistent example source program.
See Persistence Guide document.

1.7.50 Example ume-example-src-3.c

Source code: [ume-example-src-3.c](#)

ume-example-src-3.c: - Persistent example source program.
See Persistence Guide document.

1.7.51 Example ume-example-src.c

Source code: [ume-example-src.c](#)

ume-example-src.c: - Persistent example receiver program.
See Persistence Guide document.

1.7.52 Example umercv.c

Source code: [umercv.c](#)

Purpose: application that receives persisted messages from a given topic.

Usage: umercv [options] topic

Available options:

-A, --ascii	display messages as ASCII text (-A -A for newlines after each msg)
-c, --config=FILE	Use LBM configuration file FILE. Multiple config files are allowed. Example: '-c file1.cfg -c file2.cfg'
-D, --deregister=NUM	Deregister the receiver after receiving NUM messages
-E, --exit	exit after source ends
-e, --explicit-ack=N	send an Explicit ACK every N messages
-h, --help	display this help and exit
--max-sources=num	allow num sources (for statistics gathering purposes)
-i, --regid-offset=offset	use offset to calculate Registration ID (as source registration ID + offset) offset of 0 forces creation of regid by store display recovery sequence number info and set low
-N, --seqnum=X	seqnum to low+X
-r, --msgs=NUM	delete receiver after NUM messages
--session-id=NUM	Use NUM as a Session ID rather than using a Registration ID (regid-offset will be ignored)
-s, --statistics=NUM	print statistics every NUM seconds, along with bandwidth
-S, --stop	exit after source ends, print throughput summary
-u, --uregid=num	set User settable Registration ID to num for context
-v, --verbose	be verbose about incoming messages (-v -v = be even more verbose)
-V, --verify	verify message contents
-x, --no-exit-on-reg-error	don't exit on registration error (default is to exit)

1.7.53 Example umesnaprepo.c

Source code: [umesnaprepo.c](#) See [umesnaprepo Man Page](#) for usage information.

1.7.54 Example umesrc.c

Source code: [umesrc.c](#)

Purpose: application that sends persisted messages to a given topic at a specified rate.

Usage: umesrc [options] topic

Available options:

-c, --config=FILE	Use LBM configuration file FILE. Multiple config files are allowed. Example: '-c file1.cfg -c file2.cfg'
-d, --delay=NUM	delay sending for NUM seconds after source creation
-D, --deregister	deregister the source after sending messages
-h, --help	display this help and exit
-j, --late-join	turn on UME late join
-f, --flight-size=NUM	allow NUM unstabilized messages in flight (determines message rate)
-l, --length=NUM	send messages of NUM bytes
-L, --linger=NUM	linger for NUM seconds before closing context
-M, --messages=NUM	send NUM messages
-m, --message-rate=NUM	send at NUM messages per second if allowed by the flight size setting
-N, --seqnum-info	display sequence number information from source events
-n, --non-block	use non-blocking I/O
-P, --pause=NUM	pause NUM milliseconds after each send
-R, --rate=[UM]DATA/RETR	Set transport type to LBT-R[UM], set data rate limit to DATA bits per second, and set retransmit rate limit to RETR bits per second. For both limits, the optional k, m, and g suffixes may be used. For example, '-R 1m/500k' is the same as '-R 1000000/500000'
-s, --statistics=NUM	print statistics every NUM seconds
-S, --store=IP	use specified UME store
-t, --storename=NAME	use specified UME store
-v, --verbose	print additional info in verbose form
-V, --verifiable	construct verifiable messages

1.7.55 Example umessrc.c

Source code: [umessrc.c](#)

Purpose: application that uses Smart Source sends to a given topic. Understands persistence.

Usage: umessrc [options] topic

Available options:

-a, --available-data-space	print the length of available data space
-b, --user-supplied-buffer	send messages using a user-supplied buffer
-c, --config=FILE	Use LBM configuration file FILE. Multiple config files are allowed. Example: '-c file1.cfg -c file2.cfg'
-d, --delay=NUM	delay sending for NUM seconds after smart source creation
-D, --deregister	deregister the smart source after sending messages
-h, --help	display this help and exit
-i, --int-mprop=VAL,KEY	send integer message property value VAL with name KEY
-j, --late-join	turn on UME late join

-f, --flight-size=NUM	allow NUM unstabilized messages in flight (determines message rate)
-l, --length=NUM	send messages of NUM bytes
-L, --linger=NUM	linger for NUM seconds before closing context
-M, --messages=NUM	send NUM messages
-m, --message-rate=NUM	send at NUM messages per second if allowed by the flight size setting
-N, --channel=NUM	send on channel NUM
-n, --non-block	use non-blocking I/O
-P, --pause=NUM	pause NUM milliseconds after each send
-Q, --seqnum-info	display sequence number information from smart source events
-s, --statistics=NUM	print statistics every NUM seconds
-S, --store=IP	use specified UME store
-t, --storename=NAME	use specified UME store
-v, --verbose	print additional info in verbose form
-V, --verifiable	construct verifiable messages

Monitoring options:

--monitor-src=NUM	monitor source every NUM seconds
--monitor-ctx=NUM	monitor context every NUM seconds
--monitor-transport=TRANS	use monitor transport module TRANS TRANS may be 'lbm', 'lbmsnmp', or 'udp', default is 'lbm'
--monitor-transport-opts=OPTS	use OPTS as transport module options
--monitor-format=FMT	use monitor format module FMT FMT may be 'csv' or 'pb'
--monitor-format-opts=OPTS	use OPTS as format module options
--monitor-appid=ID	use ID as application ID string

Transport and format options are passed as name=value pairs, separated by a semicolon.

The entire option string should be enclosed in double-quotes.

LBM transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC default is /29west/statistics
allow_debug=VAL	VAL may be 'off' or 'on' defaults to 'off'

LBMSNMP transport options:

Note that individual LBM options can be specified as <scope>|<option>=value, where <scope> is one of context, source, receiver, wildcard_receiver, or event_queue <option> is the LBM configuration option name

The vertical bar (pipe symbol) is required when specifying individual LBM options.

config=FILE	use LBM configuration file FILE
topic=TOPIC	send statistics on topic TOPIC default is /29west/statistics

UDP transport options:

address=IP	send statistics to address IP
port=NUM	send to UDP port NUM default is 2933
mcgroup=GRP	send on multicast group GRP
bcaddress=IP	send statistics to broadcast address IP
ttl=NUM	send multicast statistics with TTL NUM default is 16

CSV format options:

separator=CHAR separate CSV fields with character CHAR
 defaults to ','
 Don't use a semicolon!

PB format options:

filters=FILE use FILE that contains filter options

1.7.56 Example umestored_example.c

Source code: [umestored_example.c](#)

umestored_example.c: application that shows how to call the umestored_main API to start a UMP store daemon.

1.7.57 Example umqrcv.c

Source code: [umqrcv.c](#)

Purpose: application that receives brokered queuing messages from a single topic.

Usage: umqrcv [options] topic

Available options:

-A, --ascii display messages as ASCII text (-A -A for newlines after each msg)
 -B, --broker=address use broker given by address
 -C, --config=FILE use FILE as LBM configuration file
 -D, --dereg deregister upon exit
 -d, --delay=NUM delay receiver creation NUM seconds from context creation
 -E, --exit exit after source ends
 -h, --help display this help and exit
 -I, --type-id=ID set Receiver Type ID to ID
 --max-sources=num allow num sources (for statistics gathering purposes)
 -r, --msgs=NUM delete receiver after NUM messages
 -s, --statistics=NUM print statistics every NUM seconds, along with bandwidth
 -S, --stop exit after source ends, print throughput summary
 -X, --index reserve given index if possible, or leave blank to reserve random index
 -v, --verbose be verbose about incoming messages
 (-v -v = be even more verbose)
 -V, --verify verify message contents

Monitoring options:

--monitor-rcv=NUM monitor receiver every NUM seconds
 --monitor-ctx=NUM monitor context every NUM seconds
 --monitor-transport=TRANS use monitor transport module TRANS
 TRANS may be 'lbm', 'lbmsnmp', or 'udp', default is 'lbm'
 --monitor-transport-opts=OPTS use OPTS as transport module options
 --monitor-format=FMT use monitor format module FMT
 FMT may be 'csv' or 'pb'
 --monitor-format-opts=OPTS use OPTS as format module options
 --monitor-appid=ID use ID as application ID string

Transport and format options are passed as name=value pairs, separated by a semicolon.

The entire option string should be enclosed in double-quotes.

LBM transport options:

Note that individual LBM options can be specified as `<scope>|<option>=value`, where `<scope>` is one of `context`, `source`, `receiver`, `wildcard_receiver`, or `event_queue` and `<option>` is the LBM configuration option name.

The vertical bar (pipe symbol) is required when specifying individual LBM options.

<code>config=FILE</code>	use LBM configuration file <code>FILE</code>
<code>topic=TOPIC</code>	send statistics on topic <code>TOPIC</code> default is <code>/29west/statistics</code>
<code>allow_debug=VAL</code>	<code>VAL</code> may be <code>'off'</code> or <code>'on'</code> defaults to <code>'off'</code>

LBMSNMP transport options:

Note that individual LBM options can be specified as `<scope>|<option>=value`, where `<scope>` is one of `context`, `source`, `receiver`, `wildcard_receiver`, or `event_queue` and `<option>` is the LBM configuration option name.

The vertical bar (pipe symbol) is required when specifying individual LBM options.

<code>config=FILE</code>	use LBM configuration file <code>FILE</code>
<code>topic=TOPIC</code>	send statistics on topic <code>TOPIC</code> default is <code>/29west/statistics</code>

UDP transport options:

<code>address=IP</code>	send statistics to address <code>IP</code>
<code>port=NUM</code>	send to UDP port <code>NUM</code> default is <code>2933</code>
<code>mcgroup=GRP</code>	send on multicast group <code>GRP</code>
<code>bcaddress=IP</code>	send statistics to broadcast address <code>IP</code>
<code>ttl=NUM</code>	send multicast statistics with TTL <code>NUM</code> default is <code>16</code>

CSV format options:

<code>separator=CHAR</code>	separate CSV fields with character <code>CHAR</code> defaults to <code>'</code> , <code>'</code> Don't use a semicolon!
-----------------------------	---

PB format options:

<code>filters=FILE</code>	use <code>FILE</code> that contains filter options
---------------------------	--

1.7.58 Example umqsrc.c

Source code: [umqsrc.c](#)

Purpose: `umqsrc.c`: application that sends brokered queuing messages to a single topic at a specified rate.

Usage: `umqsrc [options] topic`

Available options:

<code>-A, --appsets=CFG</code>	use ULB Application Sets given by <code>CFG</code>
<code>-B, --broker=address</code>	use broker given by <code>address</code>
<code>-C, --config=FILE</code>	use LBM configuration file <code>FILE</code>
<code>-d, --delay=NUM</code>	delay sending for <code>NUM</code> seconds after source creation
<code>-h, --help</code>	display this help and exit
<code>-f, --flight-size=NUM</code>	allow <code>NUM</code> unstabilized messages in flight (determines message rate)
<code>-i, --ids</code>	display Message IDs for sent message
<code>-l, --length=NUM</code>	send messages of <code>NUM</code> bytes
<code>-L, --linger=NUM</code>	linger for <code>NUM</code> seconds before closing context
<code>-M, --messages=NUM</code>	send <code>NUM</code> messages
<code>-m, --message-rate=NUM</code>	send at <code>NUM</code> messages per second
<code>-N, --seq-num</code>	display sequence number information

```

-n, --non-block           use non-blocking I/O
-P, --pause=NUM           pause NUM milliseconds after each send
-R, --rate=[UM]DATA/RETR  Set transport type to LBT-R[UM], set data rate limit to
                           DATA bits per second, and set retransmit rate limit to
                           RETR bits per second. For both limits, the optional
                           k, m, and g suffixes may be used. For example,
                           '-R 1m/500k' is the same as '-R 1000000/500000'
-s, --statistics=NUM      print statistics every NUM seconds
-v, --verbose             print additional info in verbose form
-V, --verifiable          construct verifiable messages
-X, --index               Send messages on specified index for ULB sources
-Y, --broker-index       Send messages on specified named index for broker
                           sources

```

1.8 Example Protocol Files

Google protocol buffer definition files. See **Monitoring Formats**.

1.8.1 Example dro_mon.proto

Source code: [dro_mon.proto](#)

1.8.2 Example srs_mon.proto

Source code: [srs_mon.proto](#)

1.8.3 Example um_mon_attributes.proto

Source code: [um_mon_attributes.proto](#)

1.8.4 Example um_mon_control.proto

Source code: [um_mon_control.proto](#)

1.8.5 Example ump_mon.proto

Source code: [ump_mon.proto](#)

1.8.6 Example ums_mon.proto

Source code: [ums_mon.proto](#)
