



Welcome to AMPS!

Reference card for AMPS 5.2

This card contains getting started information and a quick reference for AMPS 5.2. More comprehensive information is available on the 60East web site at crankuptheamps.com.

Learn about AMPS

AMPS includes full documentation. 60East regularly updates the documentation based on analysis of incoming questions and customer feedback. Get up-to-date documentation at

<http://crankuptheamps.com/documentation>

AMPS features and usage	<i>AMPS User Guide</i>
Configuring AMPS	<i>AMPS Configuration Guide</i>
Developing with AMPS	<i>AMPS Development Guide for Java, Python, C#/.NET, C/C++, or JavaScript</i> <i>AMPS Evaluation (downloadable and cloud versions)</i>
AMPS utilities	<i>AMPS Utilities Guide</i>

AMPS Filters

expression1 operator expression2

expression

- identifier** An XPath-like identifier for a field in the message: /id or /order/@type
- literal** A literal value: 42 or 'puppies'
- conditional** Inline comparison: IF (*expression* , *value_if_true* , *value_if_false*)
- arithmetic** Expressions combined with an arithmetic operator (+ , - , * , / , % , MOD)
- valuelist** A list of values (for use with the set operators) (2 , 4 , 6 , 8)

operator

- comparison** ==, =, >, >=, <, <=, !=, <>
- regex match** LIKE
- set** IN, NOT IN, BETWEEN
- string** BEGINS WITH, ENDS WITH

combine filters AND, OR, NOT

```
/price * /qty > 100000 AND /item = 'puppies'
/state LIKE '.*Dakota'
/id IN (5, 6, 8, 10, 11, 42)
5000 < /order/maximumBid
```

Getting Support

http://support.crankuptheamps.com/	Online support and searchable FAQ
support@crankuptheamps.com	Email support for questions
crash@crankuptheamps.com	Email address for minidump files
http://crankuptheamps.com/releases	Join the mailing list for release announcements
http://crankuptheamps.com/newsletter	Subscribe to the 60East Newsletter

When contacting support, please include:

- AMPS version number
- Operating system, distribution and kernel version
- Expected behavior, observed behavior, and differences
- If you have a minidump file, be sure to include that file in an email to crash@crankuptheamps.com.

Your company may have priority support for outages: please see the AMPS contact at your company for details.

AMPS Utilities

amps_journal_dump	Print the contents of an AMPS journal file
amps_sow_dump	Print the contents of a State of the World file
amps_upgrade	Upgrade an AMPS installation from a previous version to the current version
ampserr	Print more detailed information about an AMPS error.
spark	Command-line tool for sending commands to AMPS, used for troubleshooting and diagnostics

Working With AMPS

Start AMPS

```
$ ./bin/ampServer config_file
```

Create a sample configuration file

```
$ ./bin/ampServer --sample-config > sample.xml
```

spark Quick Reference

Check to see if an AMPS server is reachable at the specified address

```
$ ./spark ping -server localhost:9007 -proto amps -type json
```

Publish

Publish from standard input

```
$ echo '{ "id": 1, "data": "boo" }' \  
| ./spark publish -server localhost:9007 \  
-topic messages -proto amps -type json
```

Publish from a file

```
$ ./spark publish -server localhost:9007 -proto amps \  
-type json -topic messages -file saved-messages.json
```

spark publishes each line in the file as a message

Subscribe

Subscribe to a topic

```
$ ./spark subscribe -server localhost:9007 -topic messages \  
-proto amps -type json
```

Subscribe to a topic with a filter

```
$ ./spark subscribe -server localhost:9007 -proto amps \  
-type json -topic messages -filter '/id > 20'
```

spark stays running and prints messages from the subscription as the messages arrive.

Bookmark Replay

Replay messages from the beginning of the transaction log

```
$ ./spark subscribe -server localhost:9007 -proto amps \  
-type json -topic Trades -bookmark 0
```

Replay messages from a specific point in time

```
$ ./spark subscribe -server localhost:9007 -proto amps \  
-type json -topic Trades -bookmark 20170601T083000
```

spark replays the historical messages and stays running and prints messages from the subscription as new messages arrive.

SOW

Retrieve the current state of a SOW topic

```
$ ./spark sow -server localhost:9007 -proto amps \  
-type json -topic sow-test
```

Retrieve the current state of a SOW topic in order

```
$ ./spark sow -server localhost:9007 -proto amps \  
-type json -topic sow-test -orderby /agent
```

Retrieve SOW messages matching a filter

```
$ ./spark sow -server localhost:9007 -proto amps \  
-type json -topic sow-test -filter '/id % 23 = 0'
```

Copy the current contents of a topic SOW to another server

```
$ ./spark sow -server hostone:9007/amps/json -topic a \  
| ./spark publish -server hosttwo:9007/amps/json \  
-topic b
```

SOW and Subscribe

Retrieve SOW messages and subscribe to updates

```
$ ./spark sow_and_subscribe -server localhost:9007 \  
-proto amps -type json -topic messages-sow \  
-filter '/id in (1, 5, 7, 42) OR /id > 90'
```

Retrieve SOW messages, subscribe to updates, and request OOF notification

```
$ ./spark sow_and_subscribe -server localhost:9007 \  
-proto amps -type json -topic messages-sow \  
-filter "/status = 'available'" -opts 'oof'
```

spark stays running and prints messages from the subscription as the messages arrive.

Delete from SOW

Delete all messages from a SOW topic

```
$ ./spark sow_delete -server localhost:9007 -proto amps \  
-type json -topic Trades -filter '1=1'
```

Delete all messages that match a filter from a SOW topic

```
$ ./spark sow_delete -server localhost:9007 -topic Trades \  
-filter "/order/id = '3457A22E'"
```