Hazelcast Open Binary Client Protocol - Version 1.0

Protocol Version	1.0
Document Release Date	25.02.2016

Introduction

This document explains the new binary protocol Hazelcast uses to communicate with clients. This document is not a guide to implement a client that will interact with Hazelcast; rather, it specifies the wire data format for messages exchanged between a client and a Hazelcast server member node (server). Any client that wants to communicate with the Hazelcast cluster should obey the data format and communication details explained in this document.

Hazelcast client protocol is a binary protocol specification. The communication between the client and Hazelcast cluster starts when a client connects to a cluster member. The protocol is designed to be strict enough to ensure a standardization in the communication, but still flexible enough that developers may expand upon the protocol to implement custom features.

General guidelines:

- This document uses the following terms as defined by IETF RFC 2119: MUST, MUST NOT, MAY, SHOULD, SHOULD NOT.
- “Client” refers to the entity which communicates with a Hazelcast member node.
- “Server” refers to the Hazelcast member to which the client connects.

Data Format Details

Hazelcast provides a communication interface to access distributed objects through client protocol. This interface is a TCP socket listening for request messages. Currently, TCP socket communication is the only way a client can connect to the server. The client MUST connect to the port that Hazelcast is listening to for new connections. Because of this, there is no specific fixed port to which the client must connect. Protocol communication is built on message sending and receiving. Client protocol defines a simple message called "client message" for communication. It is the only data format defined by this protocol.

Before we go into the details of the communication, we define a client message in the following section.

Client Message

A client message is a transmission data unit composed of a fixed header and a variable payload data. Its main purpose is to encapsulate a unit of data to be transferred from one entity to another. It may represent a request, a response or an event response. A client message can be fragmented into multiple client messages and sent in order one-by-one. Please see the Message Framing section for details. A message is composed of a header part and a payload body as shown below:

MESSAGE HEADER		MESSAGE HEADER EXTENSION	MESSAGE BODY
20 BYTES	data-offset field, 2 bytes	0 to 2^16 -22 bytes Currently Unused	PAYLOAD
<----------DATA OFFSET DEFINES THIS SIZE--------->			

The Client Message header could be extended (using the "Message Header Extension" field) or even include custom data that you (client developers) could define. In order to achieve a variable size header, you can use the data offset field of the header to set the start of the payload.

The current version of the protocol defines the header as the first 22 bytes of the Client Message as defined in the following section.

Message Header

Message header fields uniquely identify and define the client message.

Name	Data Type	Description
Frame Size	int32	Frame data size is the total size in bytes. The minimum value of the frame size can be a fixed header size of 22 bytes: header message only, no payload.
Version	uint8	Protocol version. Current version is 1.
Flags	uint8	Flag bits. Please see the table below for an explanation of the flags.
Type	uint16	Type of the message corresponding to a unique operation of a distributed object: e.g. map.put, map.get, etc., a response, a general protocol message type, or an event.
Correlation ID	int64	This ID correlates the requests to responses. It should be unique to identify one message in the communication. This ID is used to track the request response cycle of a client operation. Server side should send response messages with the same ID as the request message. The uniqueness is per connection. Each message generating entity must generate unique IDs. If the client receives the response to a request and the request is not a multi-response request (i.e. not a request for event transmission), then the correlation ID for the request can be reused by subsequent requests.
Partition ID	int32	Target partition where this message will be processed. Negative values will go into a global execution pool and do not have a partition.
Data Offset	uint16	A message may have a payload data. Data offset value must be set to the number of bytes from the beginning of the frame to the start of the payload including the "Data Offset" field.

Flag bits are depicted in the table below. The first row shows the bit numbers in the order of transmission.Please see Client Protocol Data Types f or the description of the "Data Type" column values.

8	7	6	5	4	3	2	1
BEGIN	END	N/A	N/A	N/A	N/A	N/A	EVENT

Flag bits are one bit tags to define some features of the client message. In the current protocol version, there are three different flags as defined below:

1. Begin Flag: If this bit is set, this is the first part of the message the client sends.
2. End Flag: If this bit is set, this is the final part of the message the client sends.
3. Event Flag: If this bit is set, this message is an event response from the server.

Please note that if both BEGIN and END bits are set, it means that this message is not fragmented, i.e. it starts and ends in the same message. If neither the BEGIN nor END bit is set, then the message is an intermediate part of the entire message.

Type

The message type refers to the kind of operation the client runs on the server. A full list of types can be found in Protocol Messages.

If the server receives a message with an unsupported message type, it will return the "No such message task" error to the client. The client is guaranteed to receive only the messages listed in the Protocol Messages and the error messages.

Correlation ID

The correlation ID must be a unique unsigned 32-bit integer. The server always sends a response message with the same correlation ID as the corresponding request. The correlation ID also implies how a client registers for event updates (i.e. all subsequent event updates are sent using the same correlation ID as the original registration request correlation ID). Note that once a correlation ID is used to register for an event, it SHOULD NOT be used again unless the client unregisters (stops listening) for the event.

Partition ID

The partition ID defines the partition against which the operation is executed. The client can get the partition list of the cluster. This information tells the client which member handles which partition. The client can use this information to send the related requests to the responsible member (for the request key if it exists) directly for processing. The client gets this information from the “Get Partitions” request (see the Protocol Messages).

To determine the partition ID of an operation, compute the Murmur Hash (version 3, 32-bit, see https://en.wikipedia.org/wiki/MurmurHash and https://code.google.com/p/smhasher/wiki/MurmurHash3) of a certain byte-array (which is identified for each operation) and take the modulus of the result over the total number of partitions. The seed for the Murmur Hash SHOULD be 0x01000193. Most operations with a key parameter use the key parameter byte-array as the data for the hash calculation.

Some operations are not required to be executed on a specific partition, but can be run on any partition. For these operations, the partition ID is set to a negative value. No hash calculation is required in this case.

Message Header Extension

This value should be set to 1 for this version of the protocol. It identifies the version of the prevailing protocol. Since this is the first version of the protocol, this field has no use for now.

Message Body

The payload (the body of the message) starts from byte number "data-offset" (22 for the current header definition). The total number of bytes in the payload can be calculated as "frame-size - data-offset". The message payload can carry one of the following contents:

1. If the message is a fragmented message, i.e. if either or both of the BEGIN or END flags are not set, then this body is just a fragment of an actual message that should be merged to form the actual message before message handling.
2. If the message is a non-fragmented message, i.e. both BEGIN and END flags are set, then the body is a set of binary encoded fields specific to a message type (structure is identified by the TYPE field of the header).

Please refer to Protocol Messages for details of the encoding of each message type.

Message Framing

This protocol supports messages being split into smaller sized messages (fragments) to overcome the problem of huge messages blocking the channel. Message splitting is optionally applied depending on the implementation internals and various optimization reasons. Although message splitting is optional, merging the received frames into a complete message (if fragmented) must be supported (reassembly).

A frame size may vary from a size of 22 bytes (minimum header of a client message) to 2^31-1 bytes.

If a client message is required to be split into smaller frames, then the header part must be the same in the subsequent messages for all fragments except for the flag byte. While the body part should be split and distributed into new frames consecutively, there is no set of rules on how to split the payload.

The BEGIN flag must be set in the first frame and the END flag must be set in the last frame. Intermediate messages should have no BEGIN or END flags set.

On receiving the message with the END flag set, the frames should then be merged into a single frame and then processed.

Message Types

There are three different kinds of messages used in this protocol. These are:

1. Request Message
2. Response Message
3. Event Message

Request Message

Each distributed object defines various operations. Each operation corresponds to a well defined request message to be sent to the cluster. For each request message, the client will get a response message from the cluster. Request messages MUST be sent from the client to the server.

The request parameters are binary encoded entirely within the payload of the message.

Response Message

Once a request message is received and processed at the server side, the server may produce a response message and send it to the client. Each request message type defines possible response message types that can be sent in response. The correlation ID relates all instances of the response messages to their requests.

The payload of the message is defined by the message type. The payload includes binary encoded parameters of the response.

Event Message

An event message is a special kind of response message. A client can register to a specific listener by sending a request message with the message type of adding a listener (for the specific message types that add listeners, see the “Type” section of the header format). When an event is triggered that the client is listening for, the server will send a message to the client using the same correlation ID as the original request message. The payload of the message carries the specific event object. The possible event message types for a registration request are documented in the "Events" section of each request in the Protocol Messages document.

The server will continue to send the client event updates until the client unregisters from that event or the connection is broken.

Client Protocol Data Types

Type	Description	Size	Min Value	Max Value
uint8	unsigned 8 bit integer	8 bit	0	255 (2^8 - 1)
uint16	unsigned 16 bit integer	16 bit	0	65535 (2^16 - 1)
uint32	unsigned 32 bit integer	32 bit	0	2^32 - 1
uint64	unsigned 64 bit integer	64 bit	0	2^64 - 1
int8	signed 8 bit integer in 2's compliment	8 bit	-128	127
int16	signed 16 bit integer in 2's compliment	16 bit	-32768	32767
int32	signed 32 bit integer in 2's compliment	32 bit	-2^31	2^31 - 1
int64	signed 64 bit integer in 2's compliment	64 bit	-2^63	2^63 - 1
float	single precision IEEE 754, see link for details	32 bit		
double	double precision IEEE 754, see link for details	64 bit		
boolean	same as uint8 with special meanings 0 is "false" any other value is "true"	8 bit	N/A	N/A
string	String encoded as a byte array with UTF-8 encoding. Initial 4 bytes (uint32 type) represents the size of the string in bytes Example: the string “bad” {0x62,0x61,0x64} will be encoded as 0x03,0x00,0x00,0x00,0x62,0x61,0x64. Only UTF-8 code points 0x0000 up to 0xFFFF (3-bytes) are supported.	variable		
byte-array	Variable size bytes with a uint32 size prefix. Example: { 0x11, 0x22 } will be encoded as ,0x02, 0x00, 0x00, 0x00,0x11, 0x22	variable		

Data types are consistent with those defined in The Open Group Base Specifications Issue 7 IEEE Std 1003.1, 2013 Edition. Data types are in Little Endian format.

Connection Guide

Open Connection

As previously stated, TCP socket communication is used for client-server communication. Each server has a server socket listening for incoming connections. This TCP socket is used for both server-to-server and client-to-server communications.

As the first step of client-to-server communication, the client MUST open a TCP socket connection to the server.

A client needs to establish a single connection to each member node if it is a smart client. If it is a dummy client, a single connection is enough for the client. For more information, please see Dummy Client versus Smart Client.

Connection Initialization

After successfully connecting to the server TCP socket port, the client MUST send three bytes of initialization data to identify the connection type to the server.

For any client, the three byte initializer data is [0x43, 0x42, 0x32], which is the string "CB2" in utf8 encoding.

Authentication

The first message sent through an initialized connection must be an authentication message. Any other type of messages will fail with an authorization error unless the authentication is complete. Authentication is the process of sending an authentication message and receiving an authentication response.

Upon successful authentication, the client will receive a response from the server with the server's IP address, the connection UUID and the owner UUID. The authentication response status field should be checked for the authentication success. There are three poss'ble statuses:

1. Authentication is successful
2. Credentials failed: The provided credentials (e.g username/password is incorrect)
3. Serialization version mismatch: The requested serialization version and the serialization version used at the server are different. The client can get the server's serialization version from the serializationVersion field of the response. It is suggested that the client tries to reconnect using the matching serialization version assuming that the client implements the version for serialization.

There are two types of authentications:

1. Username/Password authentication: "Authentication" (See Protocol Messages for message fields) message is used for this authentication. The response is AuthenticationResult message. The result contains the address, UUID and owner UUID information.
2. Custom credentials authentication: “Custom Authentication” message is used. This method sends the custom authentication credentials as a byte array. The response is the same as the username/password authentication.

One of the parameters in the Authentication request is the boolean flag “isOwnerConnection”. Each client is associated with an owner server. This owner server tracks the availability of the client and if it detects that the client does not exist, it cleans up any resource allocated for that client. The locks acquired by the client is one such resource. “isOwnerConnection” needs to be set to true if this connection will be the owner connection.

Communication via Messages

After a successful authentication, a client may send request messages to the server to access distributed objects or perform other operations on the cluster. This step is the actual communication step.

Once connected, a client can do the following:

1. Send periodic updates.
2. Retrieve partition list.
3. Send operation messages and receive responses.
4. Get updates on cluster member changes.

All request messages will be sent to the server and all response and event messages will be sent to the client.

See Requests and Responses for details.

Closing Connections

To end the communication, the network socket that was opened in the "Open Connection" step should be closed. This will result in releasing resources on the server side specific to this connection.

Requests and Responses

Distributed Objects

To access distributed object information, use the “Get Distributed Object” message type (See Protocol Messages).

To add a listener for adding distributed objects, use the “Add Distributed Object Listener” message type.

To remove a listener for adding distributed objects, use the “Remove Distributed Object Listener" message type.

See Listeners for more information.

Partition List

The client can get the partition list of the cluster. This information tells the client which member handles which partition key. The client can use this information to send the related requests to the responsible member (for the request key if it exists) directly for processing. The request message is the “Get Partitions” request message, which has no payload.

The response contains the full cluster member list and the member-partition ownership information (using the owner index array).

To create a listener for the case of a partition being lost, use the “Add Partition Lost” request message. See below for the request, response, and event formatting.

To remove the listener, use the “Remove Partition Lost Listener” message.

Operation Messages And Responses

Operational messages are the messages where a client can expect exactly one response for a given request. The client knows which request the response correlates to via the correlation ID. An example of one of these messages is a “map put” operation.

General operation messages are listed in the "General Protocol Operations" section of Protocol Messages. To execute a particular operation, set the message type ID to the corresponding operation type. If a parameter is followed by "isNullable=true", this means that the parameter may be null. Such a parameter is encoded first by the boolean value being true or false, followed by the parameter value if the parameter is not null.

Proxies

Before using a distributed object, you SHOULD first create a proxy for the object. Do this by using the “Create Proxy” request message (See Protocol Messages).

To destroy a proxy, use the “Destroy Proxy” request message.

Java Example:

```
HazelcastInstance client = HazelcastClient.newHazelcastClient();


IMap map=client.getMap("map-name");
```

Python Example:

```
client=HazelcastClient()
map=client.getMap("map-name")
```

Raw bytes:

Client request

0x34 0x00 0x00 0x00 //Size of message

0x01 //Version

0xc0 //Flag

0x05 0x00 //Type

0x03 0x00 0x00 0x00 0x00 0x00 0x00 0x00 //Correlation ID

0xff 0xff 0xff 0xff //Partition ID

0x12 0x00 //Data Offset

0x08 0x00 0x00 0x00 //Size of following string:

0x6d 0x61 0x70 0x2d 0x6e 0x61 0x6d 0x65 //name: "map-name"

0x12 0x00 0x00 0x00 //Size of following string:

0x68 0x7a 0x3a 0x69 0x6d 0x70 0x6c 0x3a 0x6d 0x61 0x70 0x53 0x65 0x72 0x76 0x69 0x63 0x65 //serviceName: "hz:impl:mapService"

Server response

0x12 0x00 0x00 0x00 //Size of message

0x00 //Version

0xc0 //Flag

0x64 0x00 //Type

0x03 0x00 0x00 0x00 0x00 0x00 0x00 0x00 //Correlation ID

0xff 0xff 0xff 0xff //Partition ID

0x12 0x00 //Data Offset

For a request with a key, the client SHOULD send the request to the cluster member that houses the data for the key. A client can do this by using the partition ID. For more specific information on computing a given partition ID, see its details in the header section.

The response to a request message is always one of the following:

1. Regular response message: The response is the message as listed in the protocol specification for the specific request message type.
2. An error message: See “Error Codes” section.

We give examples of operations on various data structures below:

List

Java Example

```
IList myList=client.getList("list"); //create proxy
System.out.println(myList.get(3));
```

Python Example

```
mylist=client.getList("list") #create proxy
print mylist.get(3)
```

Raw bytes

Client request

0x1e 0x00 0x00 0x00 //Size of message

0x01 //Version

0xc0 //Flags

0x05 0x0f //Type

0x05 0x00 0x00 0x00 0x00 0x00 0x00 0x00 //Correlation ID

0x96 0x00 0x00 0x00 //Partition ID

0x12 0x00 //Data Offset

0x04 0x00 0x00 0x00 //Size of following string

0x6c 0x69 0x73 0x74 //name: "list"

0x03 0x00 0x00 0x00 //index: 3

Server response

0x20 0x00 0x00 0x00 //Size of message

0x00 //Version

0xc0 //Flags

0x69 0x00 //Type

0x05 0x00 0x00 0x00 0x00 0x00 0x00 0x00 //Correlation ID

0xff 0xff 0xff 0xff //Partition ID

0x12 0x00 //Data Offset

0x00 0x09 0x00 0x00 0x00 0xff 0xff 0xff 0xf9 0x00 0x00 0x00 0x00 0x00 0x03 //response: bytes

Lock

Java Example

```
ILock myLock=client.getLock("lock"); //create proxy
myLock.lock();
```

Python Example

```
lock=client.getLock("lock") #create proxy
lock.lock()
```

Raw bytes:

Client request

0x2a 0x00 0x00 0x00 //Size of message

0x01 //Version

0xc0 //Flag

0x05 0x07 //Type

0x05 0x00 0x00 0x00 0x00 0x00 0x00 0x00 //Correlation Id

0xda 0x00 0x00 0x00 //Partition Id

0x12 0x00 //Data Offset

0x04 0x00 0x00 0x00 //Size of following string

0x6c 0x6f 0x63 0x6b //name: "lock"

0xff 0xff 0xff 0xff //leaseTime: -1

0xff 0xff 0xff 0xff //threadId: -1

Server response:

0x12 0x00 0x00 0x00 //Size of message

0x00 //Version

0xc0 //Flag

0x64 0x00 //Type

0x05 0x00 0x00 0x00 0x00 0x00 0x00 0x00 //Correlation Id

0xff 0xff 0xff 0xff //Partition Id

0x12 0x00 //Data Offset

Map

Java Example

```
String key = "key1";
int value=54


IMap myMap = client.getMap("map"); //create proxy
myMap.put(key1,value);
```

Python Example

```
key="key1"
value=54
map=client.getMap("map") #create proxy
map.put(key,value)
```

Raw bytes:

Client request

0x4e 0x00 0x00 0x00 //Size of message

0x01 //Version

0xc0 //Flag

0x01 0x01 //Type

0x05 0x00 0x00 0x00 0x00 0x00 0x00 0x00 //Correlation ID

0x98 0x00 0x00 0x00 //Partition ID

0x16 0x00 //Data Offset

0x03 0x00 0x00 0x00 //size of following string

0x6d 0x61 0x70 //name: "map"

0x14 0x00 0x00 0x00 //size of following byte object

0xff 0xff 0xff 0xf5 0x00 0x00 0x00 0x00 0x00 0x04 0x00 0x00 0x00 0x04 0x00 0x04 0x6b 0x65 0x79 0x31 //entry: bytes

0x09 0x00 0x00 0x00 0xff 0xff 0xff 0xf9

0x00 0x00 0x00 0x00 0x36 0x01 0x00 0x00

0x00 0x00 0x00 0x00 0x00 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff //I don't know why these bytes are being sent over

Server response

0x13 0x00 0x00 0x00 //Size of message

0x00 //Version

0xc0 //Flag

0x69 0x00 //Type

0x05 0x00 0x00 0x00 0x00 0x00 0x00 0x00 //Correlation ID

0x98 0x00 0x00 0x00 //Partition ID

0x12 0x00 //Data Offset

0x01 //success: true

Queue

Java Example

```
IQueue myQueue=client.getQueue("queue"); //create proxy
System.out.println(myQueue.size());
```

Python Example

```
myQueue=client.getQueue("queue") #create proxy
print myQueue.size()
```

Raw bytes:

Client request

0x1b 0x00 0x00 0x00 //Size

0x01 //Version

0xc0 //flag

0x03 0x03 //Type

0x05 0x00 0x00 0x00 0x00 0x00 0x00 0x00 //Correlation ID

0x90 0x00 0x00 0x00 //Partition ID

0x16 0x00 //Data Offset

0x5 0x00 0x00 0x00 //Size of the following string

0x71 0x71 0x65 0x75 0x65 //Name: ”queue”

Server response

0x16 0x00 0x00 0x00 //Size of message

0x00 //Version

0xc0 //flag

0x66 0x00 //Type

0x05 0x00 0x00 0x00 0x00 0x00 0x00 0x00 //Correlation ID

0xff 0xff 0xff 0xff //Partition ID

0x12 0x00 //Data Offset

0x0a 0x00 0x00 0x00 //size: 10

Set

Java Example

```
ISet set=client.getSet("set"); //create proxy
set.clear();
```

Python Example

```
set=client.getSet("set") #create proxy
set.clear()
```

Raw Bytes

Client request:

0x19 0x00 0x00 0x00 //Size of message

0x01 //Version

0xc0 //Flags

0x09 0x06 //Type

0x05 0x00 0x00 0x00 0x00 0x00 0x00 0x00 //Correlation Id

0x96 0x00 0x00 0x00 //Partition Id

0x16 0x00 //Data Offset

0x03 0x00 0x00 0x00 //Size of following string

0x73 0x65 0x74 //"set"

Server response:

0x12 0x00 0x00 0x00 //Size of message

0x01 //Version

0xc0 //Flags

0x64 0x00 //Type

0x05 0x00 0x00 0x00 0x00 0x00 0x00 0x00 //Correlation Id

0xff 0xff 0xff 0xff //Partition Id

0x12 0x00 //Data Offset

Multiple Responses to a Single Request

The client can listen for updates on a member or when specific actions are taken on the cluster. This is managed by the Event Listener mechanism. The event messages have the EVENT flag set in the message header FLAGS field and they use the same correlation ID as used in the original registration request for all the subsequent event update messages. The registration message and possible event messages sent are described in the Events section of the message descriptions in Protocol Messages.

Get Updates On Cluster Member Changes

The client can register to listen for cluster member list updates using the “Cluster Membership Listener” request message, and thus the client can retrieve the list of member server addresses in the cluster. This request message has no payload.

The response to the request message are events. The response events can be one of the following: the full member list, a specific member add/removal to the cluster, or a member server attribute change. This information is needed if the client operates as a smart client.

Listeners

Listeners are a means to communicate multiple responses to a client. The client uses one of the listener registration messages to listen for updates at the cluster. Listeners are specific to a data structure. For example, there is a specific listener for map entries versus queue entries. To see how these listeners are explicitly encoded, see the relevant message in Protocol Messages document.

Because the same correlation ID is reused for every event response for a given request, the correlation ID MUST NOT be reused from event requests unless the client unregisters the listener.

To remove all of the listeners that the client has registered to, you can send the “Remove All Listeners” request message.

Error Codes

The server may return an error response. In this case, the payload of the server’s message will contain the error message. You may choose to provide the direct error codes to the API user (developers using the API of your client) or you may use some other technique, such as exceptions, to communicate the error to the API user. See the "Error Message" section of the Protocol Messages document for details of this message.

Timeouts and Retry

It is recommended that the client should be able to handle situations where the member may not be able to return the response in an expected time interval. Even if the response to a specific message is not received, the user may or may not retry the request. If the client retries the request, they SHOULD NOT use the same correlation ID.

If no message has been sent in the server’s heartbeat time (configured by the "hazelcast.client.max.no.heartbeat.seconds" system property), the server will automatically disconnect from the client. To prevent this from occurring, a client SHOULD submit a “ping” request to the server periodically. A ping message is only sent from the client to the server; the server does NOT perform any ping request. For more information, please see the Client Protocol Implementation Guide.

Miscellaneous

Smart Client versus Dummy Client

The client can work as a smart client or as a dummy client. In both cases, a client SHOULD calculate which partition ID is responsible for the request and put this information in the Partition ID field of the header.

- Smart client: A smart client sends the request directly to the cluster member that is responsible for the related key. In order to do this, the client determines the address of the cluster member that handles the calculated partition ID. The request message will be sent on this cluster member connection.
- Dummy client: The client sends the request to any cluster member that it is connected to, regardless of the key for the request. The cluster member will in turn redirect the request to the correct member in the cluster that handles the request for the provided key.

The biggest difference between the two types of clients is that a smart client must be connected to all of the members and must constantly update its partition tables so it knows which connection to use to submit a request. Both clients are compliant with the protocol.

Serialization

While mostly an implementation detail, serialization plays a crucial role in the protocol. In order for a client to execute an operation on the server that involves a variable data structure, such as putting some entry in a map or queue, the client must be aware of how objects are serialized and deserialized so that the client can process the bytes it receives accordingly. The server and the client should use the same serialization versions in order to communicate. The version is negotiated in the connection authentication phase (See Authentication). In general, you need to use serialization byte-array type fields in the messages as specified in the Protocol Messages document. The following are examples of such objects that must be serialized before being sent over the wire:

- Key
- Value
- Old Value
- New Value
- Callable (Executor Service)
- IFunction (Atomics)
- EntryProcessor (JCache)
- ExpiryPolicy (JCache)
- CacheConfig (JCache)
- ListenerConfig (JCache)
- Mapper (MapReduce)
- CombinerFactory (MapReduce)
- ReducerFactory (MapReduce)
- KeyValueSource (MapReduce)
- Interceptor (Map)

A client may follow Hazelcast's native serialization or it may implement its own custom serialization solution. For more information on how Hazelcast serializes its objects, see the official Reference Manual serialization section.

For all the byte-array parameters, the API user should implement the following, depending on the operation type:

1. If the operation is such that no server side de-serialization is needed for the parameter, then the user can just use any serialization and there is no need to do any implementation for the server side.
2. If the operation processing at the member server requires de-serialization of the byte-array parameter, then the user should use a Java object implementing one of the Hazelcast serializations (including portable serialization) and also implementing an interface required by the server for the object during processing of the specific operation. This Java interface is specified in the protocol documentation. Furthermore, this serializer must be registered in the serialization configuration of the member server as described in the Serialization section of the Reference Manual.

The client authentication request message contains serializationVersion field. The client and server decides the serialization version to be used using this information. Hazelcast serialization versions will be matched to provide a compatible serialization. There are two cases that can occur, these are:

1. Client may have higher serialization version then the server. In that case, client will auto configure itself during authentication to match the server serialization version.
2. Client may have lower serialization version then the server. In that case, server should be configured with the system property to downgrade the server serialization version.

Security

Most of the security is configured on the server-side in a Hazelcast cluster. A client must authenticate itself, which in turn lets the server establish an end-point for the client. The server can restrict what a client can and cannot access. Current protocol does not provide an explicit support for encryption. For more information, see the Security chapter of the Hazelcast Reference Manual.

Protocol Messages

Compound Data Types Used In The Protocol Specification

Some common compound data structures used in the protocol message specification are defined in this section.

Array

In the protocol specification, an array of a data type is frequently used. An array of a data type with n entries is encoded as shown below:

Field	Type	Nullable	Description
Length	int32	No	The length of the array.
Entry 1	provided data type	No	First entry of the array
Entry 2	provided data type	No	Second entry of the array
...	...	...	...
...	...	...	...
Entry n	provided data type	No	n'th entry of the array

Address Data Type

Field	Type	Nullable	Description
Host	string	No	The name or the IP address of the server member
Port	int32	No	The port number used for this address

Cache Event Data Type

Field	Type	Nullable	Description
Cache Event type	int32	No	The type of the event. Possible values and their meanings are: CREATED(1):An event type indicating that the cache entry was created. UPDATED(2): An event type indicating that the cache entry was updated, i.e. a previous mapping existed. REMOVED(3): An event type indicating that the cache entry was removed. EXPIRED(4): An event type indicating that the cache entry has expired. EVICTED(5): An event type indicating that the cache entry has evicted. INVALIDATED(6): An event type indicating that the cache entry has invalidated for near cache invalidation. COMPLETED(7): An event type indicating that the cache operation has completed. EXPIRATION_TIME_UPDATED(8): An event type indicating that the expiration time of cache record has been updated
Name	string	No	Name of the cache
Key	byte-array	Yes	Key of the cache data
Value	byte-array	Yes	Value of the cache data
Old Value	byte-array	Yes	Old value of the cache data if exists
isOldValueAvailable	boolean	No	True if old value exist

Distributed Object Info Data Type

Field	Type	Nullable	Description
Service Name	string	No	Name of the service for the distributed object. E.g. this is "hz:impl:cacheService" for Cache object
Name	string	No	Name of the object

Entry View Data Type

Field	Type	Nullable	Description
Key	byte-array	No	Key of the entry
Value	byte-array	No	Value of the entry
Cost	int64	No	Cost of the entry
Creation Time	int64	No	Time when the entry is created
Expiration Time	int64	No	Time when the entry will expiry
Hits	int64	No	Number of hits

Last Access Time	int64	No	Time when entry is last accessed
Last Stored Time	int64	No	Time when entry is last stored
Last Update Time	int64	No	Time when entry is last updated
Version	int64	No	Version of the entry
Eviction Criteria Number	int64	No	The number of the eviction criteria applied
ttl	int64	No	Time to live for the entry

Job Partition State Data Type

Field	Type	Nullable	Description
Owner Address	Address	No	The address of the partition owner
State value	string	No	Value of the partition state. Possible values are: "WAITING": Partition waits for being calculated. "MAPPING": Partition is in mapping phase. "REDUCING": Partition is in reducing phase (mapping may still not finished when this state is reached since there is a chunked based operation underlying). "PROCESSED": Partition is fully processed "CANCELLED": Partition calculation cancelled due to an internal exception

Member Data Type

Field	Type	Nullable	Description
Address	Address	No	Address of the member server
Uuid	string	No	Unique user id of the member server
isLiteMember	boolean	No	true if the server member is a liter server, false otherwise
attribute 1 name	string	No	Name of the attribute 1
attribute 1 value	string	No	Value of the attribute 1
attribute 2 name	string	No	Name of the attribute 2
attribute 2 value	string	No	Value of the attribute 2
...	...	...	...
...	...	...	...
attribute n name	string	No	Name of the attribute n
attribute n value	string	No	Value of the attribute n

n is the number of attributes for the server member.

Query Cache Event Data Type

Field	Type	Nullable	Description
Sequence number	int64	No	The sequence number for the event
Key	byte-array	Yes	The key for the event
New Value	byte-array	Yes	The new value for the event
Event type	int32	No	The type of the event
Partition Id	int32	No	The partition id for the event key

Transaction Id Data Type

Field	Type	Nullable	Description
Format Id	int32	No	The id of the transaction format
Global Transaction Id	byte-array	No	The global id for the transaction
Branch Qualifier	byte-array	No	The qualifier for the branch

Stack Trace Data type

Field	Type	Nullable	Description
Declaring Class	string	No	The name of the class
Method Name	string	No	The name of the method
File Name	string	Yes	The name of the java source file
Line Number	int32	No	The line number in the source code file

Error Message

Response Message Type Id: 109

Field	Type	Nullable	Description
Error Code	int32	No	The unique code identifying the error
Class Name	string	No	The class name which caused the error at the server side
Message	string	Yes	The brief description of the error
Stack Trace	array of stack-trace	No	The stack trace at the server side when the error occurred
Cause Error Code	int32	No	The error code for the actual cause of the error. If no cause exists, it is set to -1
Cause Class Name	string	Yes	The name of the class that actually cause the error

The following error codes are defined in the system:

Error Name	Error Code	Description
ARRAY_INDEX_OUT_OF_BOUNDS	1	Thrown to indicate that an array has been accessed with an illegal index. The index is eith negative or greater than or equal to the size of the array.
ARRAY_STORE	2	Thrown to indicate that an attempt has been made to store the wrong type of object into a array of objects. For example, the following code generates an ArrayStoreException: Object x[] = new String[3]; x[0] = new Integer(0);
AUTHENTICATION	3	The authentication failed.
CACHE	4	
CACHE_LOADER	5	
CACHE_NOT_EXISTS	6	This exception class is thrown while creating com.hazelcast.cache.impl.CacheRecordStor instances but the cache config does not exist on the node to create the instance on. This can happen in either of two cases: the cache's config is not yet distributed to the node, or the cache has been already destroyed. For the first option, the caller can decide to just retry the operation a couple of times since distribution is executed in a asynchronous way.
CACHE_WRITER	7	
CALLER_NOT_MEMBER	8	A Retryable Hazelcast Exception that indicates that an operation was sent by a machine which isn't member in the cluster when the operation is executed.

CANCELLATION	9	Exception indicating that the result of a value-producing task, such as a FutureTask, cann be retrieved because the task was cancelled.
CLASS_CAST	10	The class conversion (cast) failed.
CLASS_NOT_FOUND	11	The class does not exists in the loaded jars at the server member.
CONCURRENT_MODIFICATION	12	You are trying to modify a resource concurrently which is not allowed.
CONFIG_MISMATCH	13	Thrown when 2 nodes want to join, but their configuration doesn't match.
CONFIGURATION	14	Thrown when something is wrong with the server or client configuration.
DISTRIBUTED_OBJECT_DESTROYED	15	The distributed object that you are trying to access is destroyed and does not exist.
DUPLICATE_INSTANCE_NAME	16	An instance with the same name already exists in the system.
EOF	17	End of file is reached (May be for a file or a socket)
ENTRY_PROCESSOR	18	
EXECUTION	19	Thrown when attempting to retrieve the result of a task that aborted by throwing an exception.
HAZELCAST	20	General internal error of Hazelcast.
HAZELCAST_INSTANCE_NOT_ACTIVE	21	The Hazelcast server instance is not active, the server is possibly initialising.
HAZELCAST_OVERLOAD	22	Thrown when the system won't handle more load due to an overload. This exception is thrown when backpressure is enabled.
HAZELCAST_SERIALIZATION	23	Error during serialization/de-serialization of data.
IO	24	An IO error occurred.
ILLEGAL_ARGUMENT	25	
ILLEGAL_ACCESS_EXCEPTION	26	
ILLEGAL_ACCESS_ERROR	27	
ILLEGAL_MONITOR_STATE	28	When an operation on a distributed object is being attempted by a thread which did not initially own the lock on the object.
ILLEGAL_STATE	29	
ILLEGAL_THREAD_STATE	30	Thrown to indicate that a thread is not in an appropriate state for the requested operation.
INDEX_OUT_OF_BOUNDS	31	Thrown to indicate that an index of some sort (such as to a list) is out of range.
INTERRUPTED	32	
INVALID_ADDRESS	33	Thrown when given address is not valid.
INVALID_CONFIGURATION	34	An InvalidConfigurationException is thrown when there is an Invalid Configuration. Invalid Configuration can be a wrong Xml Config or logical config errors that are found at real tim
MEMBER_LEFT	35	Thrown when a member left during an invocation or execution.
NEGATIVE_ARRAY_SIZE	36	The provided size of the array can not be negative but a negative number is provided.
NO_SUCH_ELEMENT	37	The requested element does not exist in the distributed object.
NOT_SERIALIZABLE	38	The object could not be serialized
NULL_POINTER	39	The server faced a null pointer exception during the operation.
OPERATION_TIMEOUT	40	An unchecked version of java.util.concurrent.TimeoutException. Some of the Hazelcast operations may throw an *OperationTimeoutException*. Hazelcast uses OperationTimeoutException to pass TimeoutException up through interfaces that do have TimeoutException in their signatures.
PARTITION_MIGRATING	41	Thrown when an operation is executed on a partition, but that partition is currently being moved around.

QUERY	42	Error during query.
QUERY_RESULT_SIZE_EXCEEDED	43	Thrown when a query exceeds a configurable result size limit.
QUORUM	44	An exception thrown when the cluster size is below the defined threshold.
REACHED_MAX_SIZE	45	Exception thrown when a write-behind MapStore rejects to accept a new element.
REJECTED_EXECUTION	46	Exception thrown by an Executor when a task cannot be accepted for execution.
REMOTE_MAP_REDUCE	47	This is used for failed remote operations. This can happen if the get result operation fails t retrieve values for some reason.
RESPONSE_ALREADY_SENT	48	There is some kind of system error causing a response to be send multiple times for some operation.
RETRYABLE_HAZELCAST	49	The operation request can be retried.
RETRYABLE_IO	50	Indicates that an operation can be retried. E.g. if map.get is send to a partition that is currently migrating, a subclass of this exception is thrown, so the caller can deal with it (e. sending the request to the new partition owner).
RUNTIME	51	
SECURITY	52	There is a security violation.
SOCKET	53	There is an error in the underlying TCP protocol
STALE_SEQUENCE	54	Thrown when accessing an item in the Ringbuffer using a sequence that is smaller than th current head sequence. This means that the and old item is read, but it isn't available anymore in the ringbuffer.
TARGET_DISCONNECTED	55	Indicates that an operation is about to be sent to a non existing machine.
TARGET_NOT_MEMBER	56	Indicates operation is sent to a machine that isn't member of the cluster.
TIMEOUT	57	
TOPIC_OVERLOAD	58	Thrown when a publisher wants to write to a topic, but there is not sufficient storage to dea with the event. This exception is only thrown in combination with the reliable topic.
TOPOLOGY_CHANGED	59	Thrown when a topology change happens during the execution of a map reduce job and t com.hazelcast.mapreduce.TopologyChangedStrategy is set to com.hazelcast.mapreduce.TopologyChangedStrategy#CANCEL_RUNNING_OPERATIO
TRANSACTION	60	Thrown when something goes wrong while dealing with transactions and transactional data-structures.
TRANSACTION_NOT_ACTIVE	61	Thrown when an a transactional operation is executed without an active transaction.
TRANSACTION_TIMED_OUT	62	Thrown when a transaction has timed out.
URI_SYNTAX	63	
UTF_DATA_FORMAT	64	
UNSUPPORTED_OPERATION	65	The message type id for the operation request is not a recognised id.
WRONG_TARGET	66	An operation is executed on the wrong machine.
XA	67	An error occurred during an XA operation.
ACCESS_CONTROL	68	Indicates that a requested access to a system resource is denied.
LOGIN	69	
UNSUPPORTED_CALLBACK	70	Signals that a CallbackHandler does not recognize a particular Callback.
NO_DATA_MEMBER	71	Thrown when there is no data member in the cluster to assign partitions.
REPLICATED_MAP_CANT_BE_CREATED	72	Thrown when {@link com.hazelcast.core.HazelcastInstance#getReplicatedMap(String)} is invoked on a lite member..
MAX_MESSAGE_SIZE_EXCEEDED	73	* Thrown when client message size exceeds Integer.MAX_VALUE.

WAN_REPLICATION_QUEUE_FULL	74	Thrown when the wan replication queues are full.
ASSERTION_ERROR	75	Thrown to indicate that an assertion has failed.
OUT_OF_MEMORY_ERROR	76	Thrown when the Java Virtual Machine cannot allocate an object because it is out of memory, and no more memory could be made available by the garbage collector.
STACK_OVERFLOW_ERROR	77	Thrown when a stack overflow occurs because an application recurses too deeply.
NATIVE_OUT_OF_MEMORY_ERROR	78	Thrown when Hazelcast cannot allocate required native memory.

Please note that there may be error messages with an error code which is not listed in this table. The client can handle this situation differently based on the particular implementation (e.g. throw an unknown error code exception).

General Protocol Operations

Generic.Authentication

This message is idempotent.

Request Message

Type Id : 0x0002
Partition Id : -1

Name	Type	Nullable	Description
username	string	No	Name of the user for authentication.
password	string	No	Password for the user.
uuid	string	Yes	Unique string identifying the connected client uniquely. This string is generated by the owner member server on initial connection. When the client connects to a non-owner member it sets this field on the request.
ownerUuid	string	Yes	Unique string identifying the server member uniquely.
isOwnerConnection	boolean	No	You must set this field to true while connecting to the owner member, otherwise set to false.
clientType	string	No	The type of the client. E.g. JAVA, CPP, CSHARP, etc.
serializationVersion	uint8	No	client side supported version to inform server side

Response Message

Type Id : 0x006b

Returns the address, uuid and owner uuid.

Name	Type	Nullable
status	uint8	No
address	Address	Yes
uuid	string	Yes
ownerUuid	string	Yes
serializationVersion	uint8	No

Generic.AuthenticationCustom

This message is idempotent.

Request Message

Type Id : 0x0003
Partition Id : -1

Name	Type	Nullable	Description
credentials	byte-array	No	Secret byte array for authentication.
uuid	string	Yes	Unique string identifying the connected client uniquely. This string is generated by the owner member server on initial connection. When the client connects to a non-owner member it sets this field on the request.
ownerUuid	string	Yes	Unique string identifying the server member uniquely.
isOwnerConnection	boolean	No	You must set this field to true while connecting to the owner member, otherwise set to false.
clientType	string	No	The type of the client. E.g. JAVA, CPP, CSHARP, etc.
serializationVersion	uint8	No	client side supported version to inform server side

Response Message

Type Id : 0x006b

Returns the address, uuid and owner uuid.

Name	Type	Nullable
status	uint8	No
address	Address	Yes
uuid	string	Yes
ownerUuid	string	Yes
serializationVersion	uint8	No

Generic.AddMembershipListener

Request Message

Type Id : 0x0004
Partition Id : -1

Name	Type	Nullable	Description
localOnly	boolean	No	if true only master node sends events, otherwise all registered nodes send all membership changes.

Response Message

Type Id : 0x0068

Returns the registration id for the listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00c8

Name	Type	Nullable	Description
member	Member	No	Cluster member server
eventType	int32	No	Type of the event. The possible values are: 1: Member added event. 2: Member removed event.

Event Message

Type Id : 0x00c9

Name	Type	Nullable	Description
members	array of Member	No	The list of members in the cluster. It is used to retrieve the initial list in the cluster when the client registers for the membership events.

Event Message

Type Id : 0x00ca

Name	Type	Nullable	Description
uuid	string	No	Unique user id of the member serve
key	string	No	Name of the attribute changed
operationType	int32	No	Type of the change. Possible values are: 1: An attribute is added 2: An attribute is removed
value	string	Yes	Value of the attribute. This field only exist for operation type of 1, otherwise this field is not transferred at all

Generic.CreateProxy

Request Message

Type Id : 0x0005
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	The distributed object name for which the proxy is being created for.

serviceName	string	No	The name of the service. Possible service names are: "hz:impl:listService" "hz:impl:queueService" "hz:impl:setService" "hz:impl:atomicLongService" "hz:impl:atomicReferenceService" "hz:impl:countDownLatchService" "hz:impl:idGeneratorService" "hz:impl:semaphoreService" "hz:impl:executorService" "hz:impl:mapService" "hz:impl:mapReduceService" "hz:impl:multiMapService" "hz:impl:quorumService" "hz:impl:replicatedMapService" "hz:impl:ringbufferService" "hz:core:proxyService" "hz:impl:reliableTopicService" "hz:impl:topicService" "hz:core:txManagerService" "hz:impl:xaService"
target	Address	No	

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Generic.DestroyProxy

Request Message

Type Id : 0x0006
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	The distributed object name for which the proxy is being destroyed for.
serviceName	string	No	The name of the service. Possible service names are: "hz:impl:listService" "hz:impl:queueService" "hz:impl:setService" "hz:impl:atomicLongService" "hz:impl:atomicReferenceService" "hz:impl:countDownLatchService" "hz:impl:idGeneratorService" "hz:impl:semaphoreService" "hz:impl:executorService" "hz:impl:mapService" "hz:impl:mapReduceService" "hz:impl:multiMapService" "hz:impl:quorumService" "hz:impl:replicatedMapService" "hz:impl:ringbufferService" "hz:core:proxyService" "hz:impl:reliableTopicService" "hz:impl:topicService" "hz:core:txManagerService" "hz:impl:xaService"

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Generic.GetPartitions

Request Message

Type Id : 0x0008
Partition Id : -1

Header only request message, no message body exist.

Response Message

Type Id : 0x006c

The partition list for each member address.

Name	Type	Nullable
partitions	array of Address-Partition Id pair	No

Generic.RemoveAllListeners

Request Message

Type Id : 0x0009
Partition Id : -1

Header only request message, no message body exist.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Generic.AddPartitionLostListener

Request Message

Type Id : 0x000a
Partition Id : -1

Name	Type	Nullable	Description
localOnly	boolean	No	if true only node that has the partition sends the request, if false sends all partition lost events.

Response Message

Type Id : 0x0068

The listener registration id.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00ce

Name	Type	Nullable	Description
partitionId	int32	No	The lost partition id.
lostBackupCount	int32	No	The number of lost backups for the partition. O: the owner, 1: first backup, 2: second backup ...
source	Address	Yes	The address of the node that dispatches the event.

Generic.RemovePartitionLostListener

This message is idempotent.

Request Message

Type Id : 0x000b
Partition Id : -1

Name	Type	Nullable	Description
registrationId	string	No	The id assigned during the listener registration.

Response Message

Type Id : 0x0065

true if the listener existed and removed, false otherwise.

Name	Type	Nullable
response	boolean	No

Generic.GetDistributedObjects

Request Message

Type Id : 0x000c
Partition Id : -1

Header only request message, no message body exist.

Response Message

Type Id : 0x006e

An array of distributed object info in the cluster.

Name	Type	Nullable
response	array of Distributed Object Info	No

Generic.AddDistributedObjectListener

Request Message

Type Id : 0x000d
Partition Id : -1

Name	Type	Nullable	Description
localOnly	boolean	No	

Response Message

Type Id : 0x0068

The registration id for the distributed object listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cf

Name	Type	Nullable	Description
name	string	No	Name of the DistributedObject instance
serviceName	string	No	Name of the service.
eventType	string	No	Can be one of the two values: "CREATED" "DESTROYED"

Generic.RemoveDistributedObjectListener

This message is idempotent.

Request Message

Type Id : 0x000e
Partition Id : -1

Name	Type	Nullable	Description
registrationId	string	No	The id assigned during the registration.

Response Message

Type Id : 0x0065

true if the listener existed and removed, false otherwise.

Name	Type	Nullable
response	boolean	No

Generic.Ping

This message is idempotent.

Request Message

Type Id : 0x000f
Partition Id : -1

Header only request message, no message body exist.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Map

Map.Put

Puts an entry into this map with a given ttl (time to live) value.Entry will expire and get evicted after the ttl
If ttl is 0, then the entry lives forever.This method returns a clone of the previous value, not the original
(identically equal) value previously put into the map.Time resolution for TTL is seconds. The given TTL value is
rounded to the next closest second value.

Request Message

Type Id : 0x0101
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the map.
key	byte-array	No	Key for the map entry.
value	byte-array	No	Value for the map entry.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
ttl	int64	No	The duration in milliseconds after which this entry shall be deleted. O means infinite.

Response Message

Type Id : 0x0069

old value of the entry

Name	Type	Nullable
response	byte-array	Yes

Map.Get

This method returns a clone of the original value, so modifying the returned value does not change the actual
value in the map. You should put the modified value back to make changes visible to all nodes. This message is idempotent.

Request Message

Type Id : 0x0102
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the map.
key	byte-array	No	Key for the map entry.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0069

The value for the key if exists

Name	Type	Nullable
response	byte-array	Yes

Map.Remove

Removes the mapping for a key from this map if it is present (optional operation).
Returns the value to which this map previously associated the key, or null if the map contained no mapping for the key.
If this map permits null values, then a return value of null does not necessarily indicate that the map contained no mapping for the key; it's also possible that the map explicitly mapped the key to null. The map will not contain a mapping for the specified key once the
call returns.

Request Message

Type Id : 0x0103
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the map.
key	byte-array	No	Key for the map entry.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0069

Clone of the removed value, not the original (identically equal) value previously put into the map.

Name	Type	Nullable
response	byte-array	Yes

Map.Replace

Replaces the entry for a key only if currently mapped to a given value.

Request Message

Type Id : 0x0104
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the map.
key	byte-array	No	Key for the map entry.
value	byte-array	No	New value for the map entry.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0069

Clone of the previous value, not the original (identically equal) value previously put into the map.

Name	Type	Nullable
response	byte-array	Yes

Map.ReplaceIfSame

Replaces the the entry for a key only if existing values equal to the testValue

Request Message

Type Id : 0x0105
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the map.
key	byte-array	No	Key for the map entry.
testValue	byte-array	No	Test the existing value against this value to find if equal to this value.
value	byte-array	No	New value for the map entry. Only replace with this value if existing value is equal to the testValue.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0065

true if value is replaced with new one, false otherwise

Name	Type	Nullable
response	boolean	No

Map.ContainsKey

Returns true if this map contains a mapping for the specified key. This message is idempotent.

Request Message

Type Id : 0x0109
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the map.
key	byte-array	No	Key for the map entry.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0065

Returns true if the key exists, otherwise returns false.

Name	Type	Nullable
response	boolean	No

Map.ContainsValue

Returns true if this map maps one or more keys to the specified value.This operation will probably require time linear in the map size for most implementations of the Map interface. This message is idempotent.

Request Message

Type Id : 0x010a
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the map.
value	byte-array	No	Value to check if exists in the map.

Response Message

Type Id : 0x0065

Returns true if the value exists, otherwise returns false.

Name	Type	Nullable
response	boolean	No

Map.RemoveIfSame

Removes the mapping for a key from this map if existing value equal to the this value

Request Message

Type Id : 0x010b
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the map.
key	byte-array	No	Key for the map entry.
value	byte-array	No	Test the existing value against this value to find if equal to this value. Only remove the entry from the map if the value is equal to this value.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0065

Returns true if the key exists and removed, otherwise returns false.

Name	Type	Nullable
response	boolean	No

Map.Delete

Removes the mapping for a key from this map if it is present.Unlike remove(Object), this operation does not return the removed value, which avoids the serialization cost of the returned value.If the removed value will not be used, a delete operation is preferred over a remove operation for better performance. The map will not contain a mapping for the specified key once the call returns.
This method breaks the contract of EntryListener. When an entry is removed by delete(), it fires an EntryEvent with a null oldValue. Also, a listener with predicates will have null values, so only keys can be queried via predicates

Request Message

Type Id : 0x010c
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the map.
key	byte-array	No	Key for the map entry.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Map.Flush

If this map has a MapStore, this method flushes all the local dirty entries by calling MapStore.storeAll() and/or MapStore.deleteAll().

Request Message

Type Id : 0x010d
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the map.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Map.TryRemove

Tries to remove the entry with the given key from this map within the specified timeout value.
If the key is already locked by another thread and/or member, then this operation will wait the timeout
amount for acquiring the lock.

Request Message

Type Id : 0x010e
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the map.
key	byte-array	No	Key for the map entry.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
timeout	int64	No	maximum time in milliseconds to wait for acquiring the lock for the key.

Response Message

Type Id : 0x0065

Returns true if successful, otherwise returns false

Name	Type	Nullable
response	boolean	No

Map.TryPut

Tries to put the given key and value into this map within a specified timeout value. If this method returns false,
it means that the caller thread could not acquire the lock for the key within the timeout duration,
thus the put operation is not successful.

Request Message

Type Id : 0x010f
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the map.
key	byte-array	No	Key for the map entry.
value	byte-array	No	New value for the map entry.

threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
timeout	int64	No	maximum time in milliseconds to wait for acquiring the lock for the key.

Response Message

Type Id : 0x0065

Returns true if successful, otherwise returns false

Name	Type	Nullable
response	boolean	No

Map.PutTransient

Same as put except that MapStore, if defined, will not be called to store/persist the entry.
If ttl is 0, then the entry lives forever.

Request Message

Type Id : 0x0110
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the map.
key	byte-array	No	Key for the map entry.
value	byte-array	No	New value for the map entry.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
ttl	int64	No	The duration in milliseconds after which this entry shall be deleted. O means infinite.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Map.PutIfAbsent

Puts an entry into this map with a given ttl (time to live) value if the specified key is not already associated
with a value. Entry will expire and get evicted after the ttl.

Request Message

Type Id : 0x0111
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the map.
key	byte-array	No	Key for the map entry.
value	byte-array	No	New value for the map entry.

threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
ttl	int64	No	The duration in milliseconds after which this entry shall be deleted. O means infinite.

Response Message

Type Id : 0x0069

returns a clone of the previous value, not the original (identically equal) value previously put into the map.

Name	Type	Nullable
response	byte-array	Yes

Map.Set

Puts an entry into this map with a given ttl (time to live) value.Entry will expire and get evicted after the ttl
If ttl is 0, then the entry lives forever. Similar to the put operation except that set doesn't
return the old value, which is more efficient.

Request Message

Type Id : 0x0112
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the map.
key	byte-array	No	Key for the map entry.
value	byte-array	No	New value for the map entry.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
ttl	int64	No	The duration in milliseconds after which this entry shall be deleted. O means infinite.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Map.Lock

Acquires the lock for the specified lease time.After lease time, lock will be released.If the lock is not
available then the current thread becomes disabled for thread scheduling purposes and lies dormant until the lock
has been acquired.
Scope of the lock is this map only. Acquired lock is only for the key in this map. Locks are re-entrant,
so if the key is locked N times then it should be unlocked N times before another thread can acquire it.

Request Message

Type Id : 0x0113
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the map.

key	byte-array	No	Key for the map entry.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
ttl	int64	No	The duration in milliseconds after which this entry shall be deleted. O means infinite.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Map.TryLock

Tries to acquire the lock for the specified key for the specified lease time.After lease time, the lock will be released.If the lock is not available, then the current thread becomes disabled for thread scheduling purposes and lies dormant until one of two things happens the lock is acquired by the current thread, or the specified waiting time elapses.

Request Message

Type Id : 0x0114
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the map.
key	byte-array	No	Key for the map entry.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
lease	int64	No	time in milliseconds to wait before releasing the lock.
timeout	int64	No	maximum time to wait for getting the lock.

Response Message

Type Id : 0x0065

Returns true if successful, otherwise returns false

Name	Type	Nullable
response	boolean	No

Map.IsLocked

Checks the lock for the specified key.If the lock is acquired then returns true, else returns false. This message is idempotent.

Request Message

Type Id : 0x0115
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	name of map
key	byte-array	No	Key for the map entry to check if it is locked.

Response Message

Type Id : 0x0065

Returns true if the entry is locked, otherwise returns false

Name	Type	Nullable
response	boolean	No

Map.Unlock

Releases the lock for the specified key. It never blocks and returns immediately.
If the current thread is the holder of this lock, then the hold count is decremented.If the hold count is zero,
then the lock is released. If the current thread is not the holder of this lock,
then ILLEGAL_MONITOR_STATE is thrown.

Request Message

Type Id : 0x0116
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	name of map
key	byte-array	No	Key for the map entry to unlock
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Map.AddInterceptor

Adds an interceptor for this map. Added interceptor will intercept operations
and execute user defined methods and will cancel operations if user defined method throw exception.

Request Message

Type Id : 0x0117
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
interceptor	byte-array	No	interceptor to add

Response Message

Type Id : 0x0068

id of registered interceptor.

Name	Type	Nullable
response	string	No

Map.RemoveInterceptor

Removes the given interceptor for this map so it will not intercept operations anymore.

Request Message

Type Id : 0x0118
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
id	string	No	of interceptor

Response Message

Type Id : 0x0065

Returns true if successful, otherwise returns false

Name	Type	Nullable
response	boolean	No

Map.AddEntryListenerToKeyWithPredicate

Adds a MapListener for this map. To receive an event, you should implement a corresponding MapListener sub-interface for that event.

Request Message

Type Id : 0x0119
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
key	byte-array	No	Key for the map entry.
predicate	byte-array	No	predicate for filtering entries.
includeValue	boolean	No	true if EntryEvent should contain the value.
listenerFlags	int32	No	flags of enabled listeners.
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

A unique string which is used as a key to remove the listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cb

Name	Type	Nullable	Description
key	byte-array	Yes	The key for the entry in the map.
value	byte-array	Yes	The value of the entry in the map.
oldValue	byte-array	Yes	The original value for the key in the map if exists.
mergingValue	byte-array	Yes	The incoming merging value of the entry event.
eventType	int32	No	Possible types are: ADDED(1) REMOVED(2) UPDATED(3) EVICTED(4) EVICT_ALL(5) CLEAR_ALL(6) MERGED(7)
uuid	string	No	The id of the member.
numberOfAffectedEntries	int32	No	The number of entries affected in the map.

Map.AddEntryListenerWithPredicate

Adds an continuous entry listener for this map. Listener will get notified for map add/remove/update/evict events filtered by the given predicate.

Request Message

Type Id : 0x011a
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
predicate	byte-array	No	predicate for filtering entries.
includeValue	boolean	No	true if EntryEvent should contain the value.
listenerFlags	int32	No	flags of enabled listeners.
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

A unique string which is used as a key to remove the listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cb

Name	Type	Nullable	Description
key	byte-array	Yes	The key for the entry in the map.
value	byte-array	Yes	The value of the entry in the map.
oldValue	byte-array	Yes	The original value for the key in the map if exists.
mergingValue	byte-array	Yes	The incoming merging value of the entry event.
eventType	int32	No	Possible types are: ADDED(1) REMOVED(2) UPDATED(3) EVICTED(4) EVICT_ALL(5) CLEAR_ALL(6) MERGED(7)
uuid	string	No	The id of the member.
numberOfAffectedEntries	int32	No	The number of entries affected in the map.

Map.AddEntryListenerToKey

Adds a MapListener for this map. To receive an event, you should implement a corresponding MapListener sub-interface for that event.

Request Message

Type Id : 0x011b
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
key	byte-array	No	Key for the map entry.
includeValue	boolean	No	true if EntryEvent should contain the value.
listenerFlags	int32	No	flags of enabled listeners.
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

A unique string which is used as a key to remove the listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cb

Name	Type	Nullable	Description
key	byte-array	Yes	The key for the entry in the map.
value	byte-array	Yes	The value of the entry in the map.
oldValue	byte-array	Yes	The original value for the key in the map if exists.

mergingValue	byte-array	Yes	The incoming merging value of the entry event.
eventType	int32	No	Possible types are: ADDED(1) REMOVED(2) UPDATED(3) EVICTED(4) EVICT_ALL(5) CLEAR_ALL(6) MERGED(7)
uuid	string	No	The id of the member.
numberOfAffectedEntries	int32	No	The number of entries affected in the map.

Map.AddEntryListener

Adds a MapListener for this map. To receive an event, you should implement a corresponding MapListener sub-interface for that event.

Request Message

Type Id : 0x011c
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
includeValue	boolean	No	true if EntryEvent should contain the value.
listenerFlags	int32	No	flags of enabled listeners.
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

A unique string which is used as a key to remove the listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cb

Name	Type	Nullable	Description
key	byte-array	Yes	The key for the entry in the map.
value	byte-array	Yes	The value of the entry in the map.
oldValue	byte-array	Yes	The original value for the key in the map if exists.
mergingValue	byte-array	Yes	The incoming merging value of the entry event.
eventType	int32	No	Possible types are: ADDED(1) REMOVED(2) UPDATED(3) EVICTED(4) EVICT_ALL(5) CLEAR_ALL(6) MERGED(7)

uuid	string	No	The id of the member.
numberOfAffectedEntries	int32	No	The number of entries affected in the map.

Map.AddNearCacheEntryListener

Adds an entry listener for this map. Listener will get notified for all map add/remove/update/evict events.

Request Message

Type Id : 0x011d
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
listenerFlags	int32	No	flags of enabled listeners.
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

A unique string which is used as a key to remove the listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00d7

Name	Type	Nullable	Description
key	byte-array	Yes	The key for the entry.

Event Message

Type Id : 0x00d8

Name	Type	Nullable	Description
keys	array of byte-array	No	The keys for the entries in batch invalidation.

Map.RemoveEntryListener

Removes the specified entry listener. Returns silently if there is no such listener added before. This message is idempotent.

Request Message

Type Id : 0x011e
Partition Id : -1

Name	Type	Nullable	Description

name	string	No	name of map
registrationId	string	No	id of registered listener.

Response Message

Type Id : 0x0065

true if registration is removed, false otherwise.

Name	Type	Nullable
response	boolean	No

Map.AddPartitionLostListener

Adds a MapPartitionLostListener. The addPartitionLostListener returns a register-id. This id is needed to remove the MapPartitionLostListener using the removePartitionLostListener(String) method.
There is no check for duplicate registrations, so if you register the listener twice, it will get events twice.
IMPORTANT: Please see com.hazelcast.partition.PartitionLostListener for weaknesses.
IMPORTANT: Listeners registered from HazelcastClient may miss some of the map partition lost events due to design limitations.

Request Message

Type Id : 0x011f
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

returns the registration id for the MapPartitionLostListener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00d1

Name	Type	Nullable	Description
partitionId	int32	No	The partition id that has been lost for the given map
uuid	string	No	The id of the server member.

Map.RemovePartitionLostListener

Removes the specified map partition lost listener. Returns silently if there is no such listener added before. This message is idempotent.

Request Message

Type Id : 0x0120
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
registrationId	string	No	id of register

Response Message

Type Id : 0x0065

true if registration is removed, false otherwise.

Name	Type	Nullable
response	boolean	No

Map.GetEntryView

Returns the EntryView for the specified key.
This method returns a clone of original mapping, modifying the returned value does not change the actual value in the map. One should put modified value back to make changes visible to all nodes. This message is idempotent.

Request Message

Type Id : 0x0121
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	name of map
key	byte-array	No	the key of the entry.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x006f

EntryView of the specified key.

Name	Type	Nullable
response	array of Entry View	Yes

Map.Evict

Evicts the specified key from this map. If a MapStore is defined for this map, then the entry is not deleted from the underlying MapStore, evict only removes the entry from the memory.

Request Message

Type Id : 0x0122

Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	name of map
key	byte-array	No	the specified key to evict from this map.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0065

true if the key is evicted, false otherwise.

Name	Type	Nullable
response	boolean	No

Map.EvictAll

Evicts all keys from this map except the locked ones. If a MapStore is defined for this map, deleteAll is not called by this method. If you do want to deleteAll to be called use the clear method. The EVICT_ALL event is fired for any registered listeners.

Request Message

Type Id : 0x0123
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Map.LoadAll

Loads all keys into the store. This is a batch load operation so that an implementation can optimize the multiple loads.

Request Message

Type Id : 0x0124
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
replaceExistingValues	boolean	No	when `true`, existing values in the Map will be replaced by those loaded from the MapLoader

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Map.LoadGivenKeys

Loads the given keys. This is a batch load operation so that an implementation can optimize the multiple loads.

Request Message

Type Id : 0x0125
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
keys	array of byte-array	No	keys to load
replaceExistingValues	boolean	No	when `true`, existing values in the Map will be replaced by those loaded from the MapLoader

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Map.KeySet

Returns a set clone of the keys contained in this map. The set is NOT backed by the map, so changes to the map are NOT reflected in the set, and vice-versa. This method is always executed by a distributed query, so it may throw a QueryResultSizeExceededException if query result size limit is configured.

Request Message

Type Id : 0x0126
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of the map

Response Message

Type Id : 0x006a

a set clone of the keys contained in this map.

Name	Type	Nullable
response	array of byte-array	No

Map.GetAll

Returns the entries for the given keys. If any keys are not present in the Map, it will call loadAll The returned map is NOT backed by the original map, so changes to the original map are NOT reflected in the returned map, and vice-versa. Please note that all the keys in the request should belong to the partition id to which this request is being sent, all keys

matching to a different partition id shall be ignored. The API implementation using this request may need to send multiple of these request messages for filling a request for a key set if the keys belong to different partitions.

Request Message

Type Id : 0x0127
Partition Id : Murmur hash of any key belongs to target partition % partition count

Name	Type	Nullable	Description
name	string	No	name of map
keys	array of byte-array	No	keys to get

Response Message

Type Id : 0x0075

values for the provided keys.

Name	Type	Nullable
response	array of key-value byte array pair	No

Map.Values

Returns a collection clone of the values contained in this map.
The collection is NOT backed by the map, so changes to the map are NOT reflected in the collection, and vice-versa.
This method is always executed by a distributed query, so it may throw a QueryResultSizeExceededException
if query result size limit is configured.

Request Message

Type Id : 0x0128
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map

Response Message

Type Id : 0x006a

All values in the map

Name	Type	Nullable
response	array of byte-array	No

Map.EntrySet

Returns a Set clone of the mappings contained in this map.
The collection is NOT backed by the map, so changes to the map are NOT reflected in the collection, and vice-versa.
This method is always executed by a distributed query, so it may throw a QueryResultSizeExceededException
if query result size limit is configured.

Request Message

Type Id : 0x0129
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map

Response Message

Type Id : 0x0075

a set clone of the keys mappings in this map

Name	Type	Nullable
response	array of key-value byte array pair	No

Map.KeySetWithPredicate

Queries the map based on the specified predicate and returns the keys of matching entries. Specified predicate runs on all members in parallel.The set is NOT backed by the map, so changes to the map are NOT reflected in the set, and vice-versa. This method is always executed by a distributed query, so it may throw a QueryResultSizeExceededException if query result size limit is configured.

Request Message

Type Id : 0x012a
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map.
predicate	byte-array	No	specified query criteria.

Response Message

Type Id : 0x006a

result key set for the query.

Name	Type	Nullable
response	array of byte-array	No

Map.ValuesWithPredicate

Queries the map based on the specified predicate and returns the values of matching entries.Specified predicate runs on all members in parallel. The collection is NOT backed by the map, so changes to the map are NOT reflected in the collection, and vice-versa. This method is always executed by a distributed query, so it may throw a QueryResultSizeExceededException if query result size limit is configured.

Request Message

Type Id : 0x012b
Partition Id : -1

Name	Type	Nullable	Description

name	string	No	name of map
predicate	byte-array	No	specified query criteria.

Response Message

Type Id : 0x006a

result value collection of the query.

Name	Type	Nullable
response	array of byte-array	No

Map.EntriesWithPredicate

Queries the map based on the specified predicate and returns the matching entries.Specified predicate runs on all members in parallel. The collection is NOT backed by the map, so changes to the map are NOT reflected in the collection, and vice-versa. This method is always executed by a distributed query, so it may throw a QueryResultSizeExceededException if query result size limit is configured.

Request Message

Type Id : 0x012c
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
predicate	byte-array	No	specified query criteria.

Response Message

Type Id : 0x0075

result key-value entry collection of the query.

Name	Type	Nullable
response	array of key-value byte array pair	No

Map.AddIndex

Adds an index to this map for the specified entries so that queries can run faster.If you are querying your values mostly based on age and active then you should consider indexing these fields.
Index attribute should either have a getter method or be public.You should also make sure to add the indexes before adding entries to this map.
Indexing time is executed in parallel on each partition by operation threads. The Map is not blocked during this operation.The time taken in proportional to the size of the Map and the number Members.
Until the index finishes being created, any searches for the attribute will use a full Map scan, thus avoiding using a partially built index and returning incorrect results.

Request Message

Type Id : 0x012d
Partition Id : -1

Name	Type	Nullable	Description

name	string	No	name of map
attribute	string	No	index attribute of value
ordered	boolean	No	true if index should be ordered, false otherwise.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Map.Size

Returns the number of key-value mappings in this map. If the map contains more than Integer.MAX_VALUE elements, returns Integer.MAX_VALUE This message is idempotent.

Request Message

Type Id : 0x012e
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	of map

Response Message

Type Id : 0x0066

the number of key-value mappings in this map

Name	Type	Nullable
response	int32	No

Map.IsEmpty

Returns true if this map contains no key-value mappings. This message is idempotent.

Request Message

Type Id : 0x012f
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map

Response Message

Type Id : 0x0065

true if this map contains no key-value mappings

Name	Type	Nullable
response	boolean	No

Map.PutAll

Copies all of the mappings from the specified map to this map (optional operation).The effect of this call is equivalent to that of calling put(Object,Object) put(k, v) on this map once for each mapping from key k to value v in the specified map.The behavior of this operation is undefined if the specified map is modified while the operation is in progress.
Please note that all the keys in the request should belong to the partition id to which this request is being sent, all keys matching to a different partition id shall be ignored. The API implementation using this request may need to send multiple of these request messages for filling a request for a key set if the keys belong to different partitions.

Request Message

Type Id : 0x0130
Partition Id : Murmur hash of any key belongs to target partition % partition count

Name	Type	Nullable	Description
name	string	No	name of map
entries	array of key-value byte array pair	No	mappings to be stored in this map

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Map.Clear

This method clears the map and invokes MapStore#deleteAll deleteAll on MapStore which, if connected to a database, will delete the records from that database. The MAP_CLEARED event is fired for any registered listeners.
To clear a map without calling MapStore#deleteAll, use #evictAll.

Request Message

Type Id : 0x0131
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	of map

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Map.ExecuteOnKey

Applies the user defined EntryProcessor to the entry mapped by the key. Returns the the object which is result of the process() method of EntryProcessor.

Request Message

Type Id : 0x0132
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	name of map
entryProcessor	byte-array	No	processor to execute on the map entry
key	byte-array	No	the key of the map entry.
threadId	int64	No	

Response Message

Type Id : 0x0069

result of entry process.

Name	Type	Nullable
response	byte-array	Yes

Map.SubmitToKey

Applies the user defined EntryProcessor to the entry mapped by the key. Returns immediately with a Future representing that task.EntryProcessor is not cancellable, so calling Future.cancel() method won't cancel the operation of EntryProcessor.

Request Message

Type Id : 0x0133
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	name of map
entryProcessor	byte-array	No	entry processor to be executed on the entry.
key	byte-array	No	the key of the map entry.
threadId	int64	No	

Response Message

Type Id : 0x0069

result of entry process.

Name	Type	Nullable
response	byte-array	Yes

Map.ExecuteOnAllKeys

Applies the user defined EntryProcessor to the all entries in the map.Returns the results mapped by each key in the map.

Request Message

Type Id : 0x0134

Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
entryProcessor	byte-array	No	entry processor to be executed.

Response Message

Type Id : 0x0075

results of entry process on the entries

Name	Type	Nullable
response	array of key-value byte array pair	No

Map.ExecuteWithPredicate

Applies the user defined EntryProcessor to the entries in the map which satisfies provided predicate.
Returns the results mapped by each key in the map.

Request Message

Type Id : 0x0135
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
entryProcessor	byte-array	No	entry processor to be executed.
predicate	byte-array	No	specified query criteria.

Response Message

Type Id : 0x0075

results of entry process on the entries matching the query criteria

Name	Type	Nullable
response	array of key-value byte array pair	No

Map.ExecuteOnKeys

Applies the user defined EntryProcessor to the entries mapped by the collection of keys.The results mapped by each key in the collection.

Request Message

Type Id : 0x0136
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map

entryProcessor	byte-array	No	entry processor to be executed.
keys	array of byte-array	No	The keys for the entries for which the entry processor shall be executed on.

Response Message

Type Id : 0x0075

results of entry process on the entries with the provided keys

Name	Type	Nullable
response	array of key-value byte array pair	No

Map.ForceUnlock

Releases the lock for the specified key regardless of the lock owner.It always successfully unlocks the key, never blocks,and returns immediately.

Request Message

Type Id : 0x0137
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	name of map
key	byte-array	No	the key of the map entry.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Map.KeySetWithPagingPredicate

Request Message

Type Id : 0x0138
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
predicate	byte-array	No	specified query criteria.

Response Message

Type Id : 0x006a

result keys for the query.

Name	Type	Nullable
response	array of byte-array	No

Map.ValuesWithPagingPredicate

Queries the map based on the specified predicate and returns the values of matching entries. Specified predicate runs on all members in parallel. The collection is NOT backed by the map, so changes to the map are NOT reflected in the collection, and vice-versa. This method is always executed by a distributed query, so it may throw a QueryResultSizeExceededException if query result size limit is configured.

Request Message

Type Id : 0x0139
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
predicate	byte-array	No	specified query criteria.

Response Message

Type Id : 0x0075

values for the query.

Name	Type	Nullable
response	array of key-value byte array pair	No

Map.EntriesWithPagingPredicate

Request Message

Type Id : 0x013a
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of map
predicate	byte-array	No	specified query criteria.

Response Message

Type Id : 0x0075

key-value pairs for the query.

Name	Type	Nullable
response	array of key-value byte array pair	No

Map.ClearNearCache

Request Message

Type Id : 0x013b

Partition Id : -1

Name	Type	Nullable	Description
name	string	No	
target	Address	No	

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

MultiMap

MultiMap.Put

Stores a key-value pair in the multimap.

Request Message

Type Id : 0x0201
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
key	byte-array	No	The key to be stored
value	byte-array	No	The value to be stored
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0065

True if size of the multimap is increased, false if the multimap already contains the key-value pair.

Name	Type	Nullable
response	boolean	No

MultiMap.Get

Returns the collection of values associated with the key. The collection is NOT backed by the map, so changes to the map are NOT reflected in the collection, and vice-versa. This message is idempotent.

Request Message

Type Id : 0x0202

Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
key	byte-array	No	The key whose associated values are to be returned
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x006a

The collection of the values associated with the key.

Name	Type	Nullable
response	array of byte-array	No

MultiMap.Remove

Removes the given key value pair from the multimap.

Request Message

Type Id : 0x0203
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
key	byte-array	No	The key of the entry to remove
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x006a

True if the size of the multimap changed after the remove operation, false otherwise.

Name	Type	Nullable
response	array of byte-array	No

MultiMap.KeySet

Returns the set of keys in the multimap.The collection is NOT backed by the map, so changes to the map are NOT reflected in the collection, and vice-versa. This message is idempotent.

Request Message

Type Id : 0x0204
Partition Id : -1

Name	Type	Nullable	Description

name	string	No	Name of the MultiMap

Response Message

Type Id : 0x006a

The set of keys in the multimap. The returned set might be modifiable but it has no effect on the multimap.

Name	Type	Nullable
response	array of byte-array	No

MultiMap.Values

Returns the collection of values in the multimap.The collection is NOT backed by the map, so changes to the map are NOT reflected in the collection, and vice-versa. This message is idempotent.

Request Message

Type Id : 0x0205
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap

Response Message

Type Id : 0x006a

The collection of values in the multimap. the returned collection might be modifiable but it has no effect on the multimap.

Name	Type	Nullable
response	array of byte-array	No

MultiMap.EntrySet

Returns the set of key-value pairs in the multimap.The collection is NOT backed by the map, so changes to the map are NOT reflected in the collection, and vice-versa This message is idempotent.

Request Message

Type Id : 0x0206
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap

Response Message

Type Id : 0x0075

The set of key-value pairs in the multimap. The returned set might be modifiable but it has no effect on the multimap.

Name	Type	Nullable
response	array of key-value byte array pair	No

MultiMap.ContainsKey

Returns whether the multimap contains an entry with the key. This message is idempotent.

Request Message

Type Id : 0x0207
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
key	byte-array	No	The key whose existence is checked.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0065

True if the multimap contains an entry with the key, false otherwise.

Name	Type	Nullable
response	boolean	No

MultiMap.ContainsValue

Returns whether the multimap contains an entry with the value. This message is idempotent.

Request Message

Type Id : 0x0208
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
value	byte-array	No	The value whose existence is checked.

Response Message

Type Id : 0x0065

True if the multimap contains an entry with the value, false otherwise.

Name	Type	Nullable
response	boolean	No

MultiMap.ContainsEntry

Returns whether the multimap contains the given key-value pair. This message is idempotent.

Request Message

Type Id : 0x0209
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
key	byte-array	No	The key whose existence is checked.
value	byte-array	No	The value whose existence is checked.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation

Response Message

Type Id : 0x0065

True if the multimap contains the key-value pair, false otherwise.

Name	Type	Nullable
response	boolean	No

MultiMap.Size

Returns the number of key-value pairs in the multimap. This message is idempotent.

Request Message

Type Id : 0x020a
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap

Response Message

Type Id : 0x0066

The number of key-value pairs in the multimap.

Name	Type	Nullable
response	int32	No

MultiMap.Clear

Clears the multimap. Removes all key-value pairs.

Request Message

Type Id : 0x020b
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

MultiMap.ValueCount

Returns the number of values that match the given key in the multimap. This message is idempotent.

Request Message

Type Id : 0x020c
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
key	byte-array	No	The key whose values count is to be returned
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation

Response Message

Type Id : 0x0066

The number of values that match the given key in the multimap

Name	Type	Nullable
response	int32	No

MultiMap.AddEntryListenerToKey

Adds the specified entry listener for the specified key.The listener will be notified for all add/remove/update/evict events for the specified key only.

Request Message

Type Id : 0x020d
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
key	byte-array	No	The key to listen to
includeValue	boolean	No	True if EntryEvent should contain the value,false otherwise
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

Returns registration id for the entry listener

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cb

Name	Type	Nullable	Description
key	byte-array	Yes	The key for the entry in the map.
value	byte-array	Yes	The value of the entry in the map.
oldValue	byte-array	Yes	The original value for the key in the map if exists.
mergingValue	byte-array	Yes	The incoming merging value of the entry event.
eventType	int32	No	Possible types are: ADDED(1) REMOVED(2) UPDATED(3) EVICTED(4) EVICT_ALL(5) CLEAR_ALL(6) MERGED(7)
uuid	string	No	The id of the member.
numberOfAffectedEntries	int32	No	The number of entries affected in the map.

MultiMap.AddEntryListener

Adds an entry listener for this multimap. The listener will be notified for all multimap add/remove/update/evict events.

Request Message

Type Id : 0x020e
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
includeValue	boolean	No	True if EntryEvent should contain the value,false otherwise
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

Returns registration id for the entry listener

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cb

Name	Type	Nullable	Description
key	byte-array	Yes	The key for the entry in the map.
value	byte-array	Yes	The value of the entry in the map.
oldValue	byte-array	Yes	The original value for the key in the map if exists.
mergingValue	byte-array	Yes	The incoming merging value of the entry event.
eventType	int32	No	Possible types are: ADDED(1) REMOVED(2) UPDATED(3) EVICTED(4) EVICT_ALL(5) CLEAR_ALL(6) MERGED(7)
uuid	string	No	The id of the member.
numberOfAffectedEntries	int32	No	The number of entries affected in the map.

MultiMap.RemoveEntryListener

Removes the specified entry listener. Returns silently if no such listener was added before. This message is idempotent.

Request Message

Type Id : 0x020f
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
registrationId	string	No	Registration id of listener

Response Message

Type Id : 0x0065

True if registration is removed, false otherwise

Name	Type	Nullable
response	boolean	No

MultiMap.Lock

Acquires the lock for the specified key for the specified lease time. After the lease time, the lock will be released. If the lock is not available, then the current thread becomes disabled for thread scheduling purposes and lies dormant until the lock has been acquired. Scope of the lock is for this map only. The acquired lock is only for the key in this map.Locks are re-entrant, so if the key is locked N times, then it should be unlocked N times before another thread can acquire it.

Request Message

Type Id : 0x0210

Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
key	byte-array	No	The key the Lock
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation
ttl	int64	No	The duration in milliseconds after which this entry shall be deleted. O means infinite.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

MultiMap.TryLock

Tries to acquire the lock for the specified key for the specified lease time. After lease time, the lock will be released. If the lock is not available, then the current thread becomes disabled for thread scheduling purposes and lies dormant until one of two things happens:the lock is acquired by the current thread, or the specified waiting time elapses.

Request Message

Type Id : 0x0211
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
key	byte-array	No	Key to lock in this map.
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation
lease	int64	No	Time in milliseconds to wait before releasing the lock.
timeout	int64	No	Maximum time to wait for the lock.

Response Message

Type Id : 0x0065

True if the lock was acquired and false if the waiting time elapsed before the lock acquired

Name	Type	Nullable
response	boolean	No

MultiMap.IsLocked

Checks the lock for the specified key. If the lock is acquired, this method returns true, else it returns false. This message is idempotent.

Request Message

Type Id : 0x0212
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
key	byte-array	No	Key to lock to be checked.

Response Message

Type Id : 0x0065

True if the lock acquired,false otherwise

Name	Type	Nullable
response	boolean	No

MultiMap.Unlock

Releases the lock for the specified key regardless of the lock owner. It always successfully unlocks the key, never blocks and returns immediately.

Request Message

Type Id : 0x0213
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
key	byte-array	No	The key to Lock
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

MultiMap.ForceUnlock

Releases the lock for the specified key regardless of the lock owner. It always successfully unlocks the key, never blocks and returns immediately.

Request Message

Type Id : 0x0214
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
key	byte-array	No	The key to Lock

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

MultiMap.RemoveEntry

Removes all the entries with the given key. The collection is NOT backed by the map, so changes to the map are NOT reflected in the collection, and vice-versa.

Request Message

Type Id : 0x0215
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the MultiMap
key	byte-array	No	The key of the entry to remove
value	byte-array	No	The value of the entry to remove
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation

Response Message

Type Id : 0x0065

True if the size of the multimap changed after the remove operation, false otherwise.

Name	Type	Nullable
response	boolean	No

Queue

Queue.Offer

Inserts the specified element into this queue, waiting up to the specified wait time if necessary for space to become available.

Request Message

Type Id : 0x0301
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue
value	byte-array	No	The element to add
timeoutMillis	int64	No	Maximum time in milliseconds to wait for acquiring the lock for the key.

Response Message

Type Id : 0x0065

True if the element was added to this queue, else false

Name	Type	Nullable
response	boolean	No

Queue.Put

Inserts the specified element into this queue, waiting if necessary for space to become available.

Request Message

Type Id : 0x0302
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue
value	byte-array	No	The element to add

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Queue.Size

Returns the number of elements in this collection. If this collection contains more than Integer.MAX_VALUE elements, returns Integer.MAX_VALUE

Request Message

Type Id : 0x0303
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue

Response Message

Type Id : 0x0066

The number of elements in this collection

Name	Type	Nullable
response	int32	No

Queue.Remove

Retrieves and removes the head of this queue. This method differs from poll only in that it throws an exception if this queue is empty.

Request Message

Type Id : 0x0304
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue
value	byte-array	No	Element to be removed from this queue, if present

Response Message

Type Id : 0x0065

true if this queue changed as a result of the call

Name	Type	Nullable
response	boolean	No

Queue.Poll

Retrieves and removes the head of this queue, waiting up to the specified wait time if necessary for an element to become available.

Request Message

Type Id : 0x0305
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue
timeoutMillis	int64	No	Maximum time in milliseconds to wait for acquiring the lock for the key.

Response Message

Type Id : 0x0069

The head of this queue, or null if this queue is empty

Name	Type	Nullable
response	byte-array	Yes

Queue.Take

Retrieves and removes the head of this queue, waiting if necessary until an element becomes available.

Request Message

Type Id : 0x0306

Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue

Response Message

Type Id : 0x0069

The head of this queue

Name	Type	Nullable
response	byte-array	Yes

Queue.Peek

Retrieves, but does not remove, the head of this queue, or returns null if this queue is empty.

Request Message

Type Id : 0x0307
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue

Response Message

Type Id : 0x0069

The head of this queue, or null if this queue is empty

Name	Type	Nullable
response	byte-array	Yes

Queue.Iterator

Returns an iterator over the elements in this collection. There are no guarantees concerning the order in which the elements are returned (unless this collection is an instance of some class that provides a guarantee).

Request Message

Type Id : 0x0308
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue

Response Message

Type Id : 0x006a

list of all data in queue

Name	Type	Nullable
response	array of byte-array	No

Queue.DrainTo

Removes all available elements from this queue and adds them to the given collection. This operation may be more efficient than repeatedly polling this queue. A failure encountered while attempting to add elements to collection c may result in elements being in neither, either or both collections when the associated exception is thrown. Attempts to drain a queue to itself result in ILLEGAL_ARGUMENT. Further, the behavior of this operation is undefined if the specified collection is modified while the operation is in progress.

Request Message

Type Id : 0x0309
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue

Response Message

Type Id : 0x006a

list of all removed data in queue

Name	Type	Nullable
response	array of byte-array	No

Queue.DrainToMaxSize

Removes at most the given number of available elements from this queue and adds them to the given collection. A failure encountered while attempting to add elements to collection may result in elements being in neither, either or both collections when the associated exception is thrown. Attempts to drain a queue to itself result in ILLEGAL_ARGUMENT. Further, the behavior of this operation is undefined if the specified collection is modified while the operation is in progress.

Request Message

Type Id : 0x030a
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue
maxSize	int32	No	The maximum number of elements to transfer

Response Message

Type Id : 0x006a

list of all removed data in result of this method

Name	Type	Nullable

response	array of byte-array	No

Queue.Contains

Returns true if this queue contains the specified element. More formally, returns true if and only if this queue contains at least one element e such that value.equals(e)

Request Message

Type Id : 0x030b
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue
value	byte-array	No	Element whose presence in this collection is to be tested

Response Message

Type Id : 0x0065

true if this collection contains the specified element

Name	Type	Nullable
response	boolean	No

Queue.ContainsAll

Return true if this collection contains all of the elements in the specified collection.

Request Message

Type Id : 0x030c
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue
dataList	array of byte-array	No	Collection to be checked for containment in this collection

Response Message

Type Id : 0x0065

true if this collection contains all of the elements in the specified collection

Name	Type	Nullable
response	boolean	No

Queue.CompareAndRemoveAll

Removes all of this collection's elements that are also contained in the specified collection (optional operation).

After this call returns, this collection will contain no elements in common with the specified collection.

Request Message

Type Id : 0x030d
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue
dataList	array of byte-array	No	Collection containing elements to be removed from this collection

Response Message

Type Id : 0x0065

true if this collection changed as a result of the call

Name	Type	Nullable
response	boolean	No

Queue.CompareAndRetainAll

Retains only the elements in this collection that are contained in the specified collection (optional operation).
In other words, removes from this collection all of its elements that are not contained in the specified collection.

Request Message

Type Id : 0x030e
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue
dataList	array of byte-array	No	collection containing elements to be retained in this collection

Response Message

Type Id : 0x0065

true if this collection changed as a result of the call

Name	Type	Nullable
response	boolean	No

Queue.Clear

Removes all of the elements from this collection (optional operation). The collection will be empty after this method returns.

Request Message

Type Id : 0x030f
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Queue.AddAll

Adds all of the elements in the specified collection to this collection (optional operation).The behavior of this operation is undefined if the specified collection is modified while the operation is in progress.
(This implies that the behavior of this call is undefined if the specified collection is this collection, and this collection is nonempty.)

Request Message

Type Id : 0x0310
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue
dataList	array of byte-array	No	Collection containing elements to be added to this collection

Response Message

Type Id : 0x0065

true if this collection changed as a result of the call

Name	Type	Nullable
response	boolean	No

Queue.AddListener

Adds an listener for this collection. Listener will be notified or all collection add/remove events.

Request Message

Type Id : 0x0311
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Queue
includeValue	boolean	No	true if the updated item should be passed to the item listener, false otherwise.
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

The registration id

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cc

Name	Type	Nullable	Description
item	byte-array	Yes	Map data item.
uuid	string	No	The id of the server member.
eventType	int32	No	There are two possible values: 1: ADDED 2: REMOVED

Queue.RemoveListener

Removes the specified item listener.Returns silently if the specified listener was not added before. This message is idempotent.

Request Message

Type Id : 0x0312
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Queue
registrationId	string	No	Id of the listener registration.

Response Message

Type Id : 0x0065

True if the item listener is removed, false otherwise

Name	Type	Nullable
response	boolean	No

Queue.RemainingCapacity

Returns the number of additional elements that this queue can ideally (in the absence of memory or resource constraints) accept without blocking, or Integer.MAX_VALUE if there is no intrinsic limit. Note that you cannot always tell if an attempt to insert an element will succeed by inspecting remainingCapacity because it may be the case that another thread is about to insert or remove an element.

Request Message

Type Id : 0x0313
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue

Response Message

Type Id : 0x0066

The remaining capacity

Name	Type	Nullable
response	int32	No

Queue.IsEmpty

Returns true if this collection contains no elements.

Request Message

Type Id : 0x0314
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Queue

Response Message

Type Id : 0x0065

True if this collection contains no elements

Name	Type	Nullable
response	boolean	No

Topic

Topic.Publish

Publishes the message to all subscribers of this topic

Request Message

Type Id : 0x0401
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Topic
message	byte-array	No	The message to publish to all subscribers of this topic

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Topic.AddMessageListener

Subscribes to this topic. When someone publishes a message on this topic. onMessage() function of the given MessageListener is called. More than one message listener can be added on one instance.

Request Message

Type Id : 0x0402
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Topic
localOnly	boolean	No	if true listens only local events on registered member

Response Message

Type Id : 0x0068

returns the registration id

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cd

Name	Type	Nullable	Description
item	byte-array	No	The published item
publishTime	int64	No	The time the topic is published.
uuid	string	No	The id of the server member.

Topic.RemoveMessageListener

Stops receiving messages for the given message listener.If the given listener already removed, this method does nothing. This message is idempotent.

Request Message

Type Id : 0x0403
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Topic
registrationId	string	No	Id of listener registration.

Response Message

Type Id : 0x0065

True if registration is removed, false otherwise

Name	Type	Nullable
response	boolean	No

List

List.Size

Returns the number of elements in this list. If this list contains more than Integer.MAX_VALUE elements, returns Integer.MAX_VALUE. This message is idempotent.

Request Message

Type Id : 0x0501
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of List

Response Message

Type Id : 0x0066

The number of elements in this list

Name	Type	Nullable
response	int32	No

List.Contains

Returns true if this list contains the specified element. This message is idempotent.

Request Message

Type Id : 0x0502
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
value	byte-array	No	Element whose presence in this list is to be tested

Response Message

Type Id : 0x0065

True if this list contains the specified element, false otherwise

Name	Type	Nullable
response	boolean	No

List.ContainsAll

Returns true if this list contains all of the elements of the specified collection. This message is idempotent.

Request Message

Type Id : 0x0503
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
values	array of byte-array	No	Collection to be checked for containment in this list

Response Message

Type Id : 0x0065

True if this list contains all of the elements of the
specified collection

Name	Type	Nullable
response	boolean	No

List.Add

Appends the specified element to the end of this list (optional operation). Lists that support this operation may place limitations on what elements may be added to this list. In particular, some lists will refuse to add null elements, and others will impose restrictions on the type of elements that may be added. List classes should clearly specify in their documentation any restrictions on what elements may be added.

Request Message

Type Id : 0x0504
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
value	byte-array	No	Element to be appended to this list

Response Message

Type Id : 0x0065

true if this list changed as a result of the call, false otherwise

Name	Type	Nullable
response	boolean	No

List.Remove

Removes the first occurrence of the specified element from this list, if it is present (optional operation).
If this list does not contain the element, it is unchanged.
Returns true if this list contained the specified element (or equivalently, if this list changed as a result of the call).

Request Message

Type Id : 0x0505
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
value	byte-array	No	Element to be removed from this list, if present

Response Message

Type Id : 0x0065

True if this list contained the specified element, false otherwise

Name	Type	Nullable
response	boolean	No

List.AddAll

Appends all of the elements in the specified collection to the end of this list, in the order that they are
returned by the specified collection's iterator (optional operation).
The behavior of this operation is undefined if the specified collection is modified while the operation is in progress.
(Note that this will occur if the specified collection is this list, and it's nonempty.)

Request Message

Type Id : 0x0506
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
valueList	array of byte-array	No	Collection containing elements to be added to this list

Response Message

Type Id : 0x0065

True if this list changed as a result of the call, false otherwise

Name	Type	Nullable
response	boolean	No

List.CompareAndRemoveAll

Removes from this list all of its elements that are contained in the specified collection (optional operation).

Request Message

Type Id : 0x0507
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
values	array of byte-array	No	The list of values to compare for removal.

Response Message

Type Id : 0x0065

True if removed at least one of the items, false otherwise.

Name	Type	Nullable
response	boolean	No

List.CompareAndRetainAll

Retains only the elements in this list that are contained in the specified collection (optional operation).
In other words, removes from this list all of its elements that are not contained in the specified collection.

Request Message

Type Id : 0x0508
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
values	array of byte-array	No	The list of values to compare for retaining.

Response Message

Type Id : 0x0065

True if this list changed as a result of the call, false otherwise.

Name	Type	Nullable
response	boolean	No

List.Clear

Removes all of the elements from this list (optional operation). The list will be empty after this call returns.

Request Message

Type Id : 0x0509
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

List.GetAll

Return the all elements of this collection This message is idempotent.

Request Message

Type Id : 0x050a
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List

Response Message

Type Id : 0x006a

An array of all item values in the list.

Name	Type	Nullable
response	array of byte-array	No

List.AddListener

Adds an item listener for this collection. Listener will be notified for all collection add/remove events.

Request Message

Type Id : 0x050b
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
includeValue	boolean	No	Set to true if you want the event to contain the value.
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

Registration id for the listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cc

Name	Type	Nullable	Description
item	byte-array	Yes	Map data item.
uuid	string	No	The id of the server member.
eventType	int32	No	There are two possible values: 1: ADDED 2: REMOVED

List.RemoveListener

Removes the specified item listener. Returns silently if the specified listener was not added before. This message is idempotent.

Request Message

Type Id : 0x050c
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
registrationId	string	No	The id of the listener which was provided during registration.

Response Message

Type Id : 0x0065

True if unregistered, false otherwise.

Name	Type	Nullable
response	boolean	No

List.IsEmpty

Returns true if this list contains no elements This message is idempotent.

Request Message

Type Id : 0x050d
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List

Response Message

Type Id : 0x0065

True if this list contains no elements

Name	Type	Nullable
response	boolean	No

List.AddAllWithIndex

Inserts all of the elements in the specified collection into this list at the specified position (optional operation).
Shifts the element currently at that position (if any) and any subsequent elements to the right (increases their indices).
The new elements will appear in this list in the order that they are returned by the specified collection's iterator.
The behavior of this operation is undefined if the specified collection is modified while the operation is in progress.
(Note that this will occur if the specified collection is this list, and it's nonempty.)

Request Message

Type Id : 0x050e
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
index	int32	No	index at which to insert the first element from the specified collection.
valueList	array of byte-array	No	The list of value to insert into the list.

Response Message

Type Id : 0x0065

True if this list changed as a result of the call, false otherwise.

Name	Type	Nullable
response	boolean	No

List.Get

Returns the element at the specified position in this list This message is idempotent.

Request Message

Type Id : 0x050f
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
index	int32	No	Index of the element to return

Response Message

Type Id : 0x0069

The element at the specified position in this list

Name	Type	Nullable
response	byte-array	Yes

List.Set

The element previously at the specified position

Request Message

Type Id : 0x0510
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
index	int32	No	Index of the element to replace
value	byte-array	No	Element to be stored at the specified position

Response Message

Type Id : 0x0069

The element previously at the specified position

Name	Type	Nullable
response	byte-array	Yes

List.AddWithIndex

Inserts the specified element at the specified position in this list (optional operation). Shifts the element currently at that position (if any) and any subsequent elements to the right (adds one to their indices).

Request Message

Type Id : 0x0511
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
index	int32	No	index at which the specified element is to be inserted
value	byte-array	No	Value to be inserted.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

List.RemoveWithIndex

Removes the element at the specified position in this list (optional operation). Shifts any subsequent elements to the left (subtracts one from their indices). Returns the element that was removed from the list.

Request Message

Type Id : 0x0512
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
index	int32	No	The index of the element to be removed

Response Message

Type Id : 0x0069

The element previously at the specified position

Name	Type	Nullable
response	byte-array	Yes

List.LastIndexOf

Returns the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the element. This message is idempotent.

Request Message

Type Id : 0x0513
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
value	byte-array	No	Element to search for

Response Message

Type Id : 0x0066

the index of the last occurrence of the specified element in
this list, or -1 if this list does not contain the element

Name	Type	Nullable
response	int32	No

List.IndexOf

Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element. This message is idempotent.

Request Message

Type Id : 0x0514
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
value	byte-array	No	Element to search for

Response Message

Type Id : 0x0066

The index of the first occurrence of the specified element in
this list, or -1 if this list does not contain the element

Name	Type	Nullable
response	int32	No

List.Sub

Returns a view of the portion of this list between the specified from, inclusive, and to, exclusive.(If from and
to are equal, the returned list is empty.) The returned list is backed by this list, so non-structural changes in
the returned list are reflected in this list, and vice-versa. The returned list supports all of the optional list
operations supported by this list.
This method eliminates the need for explicit range operations (of the sort that commonly exist for arrays).
Any operation that expects a list can be used as a range operation by passing a subList view instead of a whole list.
Similar idioms may be constructed for indexOf and lastIndexOf, and all of the algorithms in the Collections class
can be applied to a subList.
The semantics of the list returned by this method become undefined if the backing list (i.e., this list) is
structurally modified in any way other than via the returned list.(Structural modifications are those that change
the size of this list, or otherwise perturb it in such a fashion that iterations in progress may yield incorrect results.) This message is idempotent.

Request Message

Type Id : 0x0515
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
from	int32	No	Low endpoint (inclusive) of the subList
to	int32	No	High endpoint (exclusive) of the subList

Response Message

Type Id : 0x006a

A view of the specified range within this list

Name	Type	Nullable
response	array of byte-array	No

List.Iterator

Returns an iterator over the elements in this list in proper sequence. This message is idempotent.

Request Message

Type Id : 0x0516
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List

Response Message

Type Id : 0x006a

An iterator over the elements in this list in proper sequence

Name	Type	Nullable
response	array of byte-array	No

List.ListIterator

Returns a list iterator over the elements in this list (in proper sequence), starting at the specified position in the list. The specified index indicates the first element that would be returned by an initial call to ListIterator#next next. An initial call to ListIterator#previous previous would return the element with the specified index minus one. This message is idempotent.

Request Message

Type Id : 0x0517
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the List
index	int32	No	index of the first element to be returned from the list iterator next

Response Message

Type Id : 0x006a

a list iterator over the elements in this list (in proper sequence), starting at the specified position in the list

Name	Type	Nullable
response	array of byte-array	No

Set

Set.Size

Returns the number of elements in this set (its cardinality). If this set contains more than Integer.MAX_VALUE elements, returns Integer.MAX_VALUE.

Request Message

Type Id : 0x0601
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Set

Response Message

Type Id : 0x0066

The number of elements in this set (its cardinality)

Name	Type	Nullable
response	int32	No

Set.Contains

Returns true if this set contains the specified element.

Request Message

Type Id : 0x0602
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Set
value	byte-array	No	Element whose presence in this set is to be tested

Response Message

Type Id : 0x0065

True if this set contains the specified element, false otherwise

Name	Type	Nullable
response	boolean	No

Set.ContainsAll

Returns true if this set contains all of the elements of the specified collection. If the specified collection is also a set, this method returns true if it is a subset of this set.

Request Message

Type Id : 0x0603
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Set
items	array of byte-array	No	Collection to be checked for containment in this list

Response Message

Type Id : 0x0065

true if this set contains all of the elements of the
specified collection

Name	Type	Nullable
response	boolean	No

Set.Add

Adds the specified element to this set if it is not already present (optional operation).
If this set already contains the element, the call leaves the set unchanged and returns false.In combination with the restriction on constructors, this ensures that sets never contain duplicate elements.
The stipulation above does not imply that sets must accept all elements; sets may refuse to add any particular element, including null, and throw an exception, as described in the specification for Collection
Individual set implementations should clearly document any restrictions on the elements that they may contain.

Request Message

Type Id : 0x0604
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Set
value	byte-array	No	Element to be added to this set

Response Message

Type Id : 0x0065

True if this set did not already contain the specified
element and the element is added, returns false otherwise.

Name	Type	Nullable
response	boolean	No

Set.Remove

Removes the specified element from this set if it is present (optional operation).
Returns true if this set contained the element (or equivalently, if this set changed as a result of the call).
(This set will not contain the element once the call returns.)

Request Message

Type Id : 0x0605
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Set
value	byte-array	No	Object to be removed from this set, if present

Response Message

Type Id : 0x0065

True if this set contained the specified element and it is removed successfully

Name	Type	Nullable
response	boolean	No

Set.AddAll

Adds all of the elements in the specified collection to this set if they're not already present (optional operation). If the specified collection is also a set, the addAll operation effectively modifies this set so that its value is the union of the two sets. The behavior of this operation is undefined if the specified collection is modified while the operation is in progress.

Request Message

Type Id : 0x0606
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Set
valueList	array of byte-array	No	Collection containing elements to be added to this set

Response Message

Type Id : 0x0065

True if this set changed as a result of the call

Name	Type	Nullable
response	boolean	No

Set.CompareAndRemoveAll

Removes from this set all of its elements that are contained in the specified collection (optional operation). If the specified collection is also a set, this operation effectively modifies this set so that its value is the asymmetric set difference of the two sets.

Request Message

Type Id : 0x0607
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Set
values	array of byte-array	No	The list of values to test for matching the item to remove.

Response Message

Type Id : 0x0065

true if at least one item in values existed and removed, false otherwise.

Name	Type	Nullable
response	boolean	No

Set.CompareAndRetainAll

Retains only the elements in this set that are contained in the specified collection (optional operation). In other words, removes from this set all of its elements that are not contained in the specified collection. If the specified collection is also a set, this operation effectively modifies this set so that its value is the intersection of the two sets.

Request Message

Type Id : 0x0608
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Set
values	array of byte-array	No	The list of values to test for matching the item to retain.

Response Message

Type Id : 0x0065

true if at least one item in values existed and it is retained, false otherwise. All items not in valueSet but in the Set are removed.

Name	Type	Nullable
response	boolean	No

Set.Clear

Removes all of the elements from this set (optional operation). The set will be empty after this call returns.

Request Message

Type Id : 0x0609
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Set

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Set.GetAll

Return the all elements of this collection

Request Message

Type Id : 0x060a
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Set

Response Message

Type Id : 0x006a

Array of all values in the Set

Name	Type	Nullable
response	array of byte-array	No

Set.AddListener

Adds an item listener for this collection. Listener will be notified for all collection add/remove events.

Request Message

Type Id : 0x060b
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Set
includeValue	boolean	No	if set to true, the event shall also include the value.
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

The registration id.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cc

Name	Type	Nullable	Description
item	byte-array	Yes	Map data item.

uuid	string	No	The id of the server member.
eventType	int32	No	There are two possible values: 1: ADDED 2: REMOVED

Set.RemoveListener

Removes the specified item listener. Returns silently if the specified listener was not added before. This message is idempotent.

Request Message

Type Id : 0x060c
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Set
registrationId	string	No	The id retrieved during registration.

Response Message

Type Id : 0x0065

true if the listener with the provided id existed and removed, false otherwise.

Name	Type	Nullable
response	boolean	No

Set.IsEmpty

Returns true if this set contains no elements.

Request Message

Type Id : 0x060d
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Set

Response Message

Type Id : 0x0065

True if this set contains no elements

Name	Type	Nullable
response	boolean	No

Lock

Lock.IsLocked

Returns whether this lock is locked or not. This message is idempotent.

Request Message

Type Id : 0x0701
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Lock

Response Message

Type Id : 0x0065

True if this lock is locked, false otherwise.

Name	Type	Nullable
response	boolean	No

Lock.IsLockedByCurrentThread

Returns whether this lock is locked by current thread or not. This message is idempotent.

Request Message

Type Id : 0x0702
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Lock
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0065

True if this lock is locked by current thread, false otherwise.

Name	Type	Nullable
response	boolean	No

Lock.GetLockCount

Returns re-entrant lock hold count, regardless of lock ownership. This message is idempotent.

Request Message

Type Id : 0x0703
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Lock

Response Message

Type Id : 0x0066

The lock hold count.

Name	Type	Nullable
response	int32	No

Lock.GetRemainingLeaseTime

Returns remaining lease time in milliseconds. If the lock is not locked then -1 will be returned This message is idempotent.

Request Message

Type Id : 0x0704
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Lock

Response Message

Type Id : 0x0067

Remaining lease time in milliseconds.

Name	Type	Nullable
response	int64	No

Lock.Lock

Acquires the lock for the specified lease time.After lease time, lock will be released.If the lock is not
available then the current thread becomes disabled for thread scheduling purposes and lies dormant until the lock
has been acquired.

Request Message

Type Id : 0x0705
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Lock
leaseTime	int64	No	Time to wait before releasing to lock

threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Lock.Unlock

Releases the lock.

Request Message

Type Id : 0x0706
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Lock
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Lock.ForceUnlock

Releases the lock regardless of the lock owner. It always successfully unlocks, never blocks,
and returns immediately.

Request Message

Type Id : 0x0707
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Lock

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Lock.TryLock

Tries to acquire the lock for the specified lease time.After lease time, the lock will be released.
If the lock is not available, then the current thread becomes disabled for thread scheduling purposes and lies
dormant until one of two things happens: the lock is acquired by the current thread, or the specified waiting

time elapses.

Request Message

Type Id : 0x0708
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Lock
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
lease	int64	No	time in milliseconds to wait before releasing the lock.
timeout	int64	No	Maximum time to wait for the lock.

Response Message

Type Id : 0x0065

true if the lock was acquired and false if the waiting time elapsed before the lock was acquired.

Name	Type	Nullable
response	boolean	No

Condition

Condition.Await

Causes the current thread to wait until it is signalled or interrupted, or the specified waiting time elapses.

Request Message

Type Id : 0x0801
Partition Id : Murmur hash of lockName % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Condition
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
timeout	int64	No	The maximum time to wait
lockName	string	No	Name of the lock to wait on.

Response Message

Type Id : 0x0065

False if the waiting time detectably elapsed before return from the method, else true

Name	Type	Nullable

response	boolean	No

Condition.BeforeAwait

Causes the current thread to wait until it is signalled or Thread#interrupt interrupted. The lock associated with this Condition is atomically released and the current thread becomes disabled for thread scheduling purposes and lies dormant until one of four things happens:
Some other thread invokes the #signal method for this Condition and the current thread happens to be chosen as the thread to be awakened; or Some other thread invokes the #signalAll method for this Condition; or Some other thread Thread#interrupt interrupts the current thread, and interruption of thread suspension is supported; or A spurious wakeup occurs.
In all cases, before this method can return the current thread must re-acquire the lock associated with this condition. When the thread returns it is guaranteed to hold this lock. If the current thread: has its interrupted status set on entry to this method; or is Thread#interrupt interrupted while waiting and interruption of thread suspension is supported, then INTERRUPTED is thrown and the current thread's interrupted status is cleared. It is not specified, in the first case, whether or not the test for interruption occurs before the lock is released.
The current thread is assumed to hold the lock associated with this Condition when this method is called. It is up to the implementation to determine if this is the case and if not, how to respond. Typically, an exception will be thrown (such as ILLEGAL_MONITOR_STATE) and the implementation must document that fact.
An implementation can favor responding to an interrupt over normal method return in response to a signal. In that case the implementation must ensure that the signal is redirected to another waiting thread, if there is one.

Request Message

Type Id : 0x0802
Partition Id : Murmur hash of lockName % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Condition
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
lockName	string	No	Name of the lock to wait on.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Condition.Signal

If any threads are waiting on this condition then one is selected for waking up.That thread must then re-acquire the lock before returning from await. An implementation may (and typically does) require that the current thread hold the lock associated with this Condition when this method is called. Implementations must document this precondition and any actions taken if the lock is not held. Typically, an exception such as ILLEGAL_MONITOR_STATE will be thrown.

Request Message

Type Id : 0x0803
Partition Id : Murmur hash of lockName % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Condition
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
lockName	string	No	Name of the lock to wait on.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Condition.SignalAll

If any threads are waiting on this condition then they are all woken up. Each thread must re-acquire the lock
before it can return from

Request Message

Type Id : 0x0804
Partition Id : Murmur hash of lockName % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Condition
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
lockName	string	No	Name of the lock to wait on.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

ExecutorService

ExecutorService.Shutdown

Initiates an orderly shutdown in which previously submitted tasks are executed, but no new tasks will be accepted.
Invocation has no additional effect if already shut down.

Request Message

Type Id : 0x0901
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the executor.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

ExecutorService.IsShutdown

Returns true if this executor has been shut down.

Request Message

Type Id : 0x0902
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the executor.

Response Message

Type Id : 0x0065

true if this executor has been shut down

Name	Type	Nullable
response	boolean	No

ExecutorService.CancelOnPartition

Request Message

Type Id : 0x0903
Partition Id : the value passed in partitionId field

Name	Type	Nullable	Description
uuid	string	No	Unique id for the execution.
partitionId	int32	No	The id of the partition to execute this cancellation request.
interrupt	boolean	No	If true, then the thread interrupt call can be used to cancel the thread, otherwise interrupt can not be used.

Response Message

Type Id : 0x0065

True if cancelled successfully, false otherwise.

Name	Type	Nullable
response	boolean	No

ExecutorService.CancelOnAddress

Request Message

Type Id : 0x0904
Partition Id : -1

Name	Type	Nullable	Description

uuid	string	No	Unique id for the execution.
address	Address	No	Address of the host to execute the request on.
interrupt	boolean	No	If true, then the thread interrupt call can be used to cancel the thread, otherwise interrupt can not be used.

Response Message

Type Id : 0x0065

True if cancelled successfully, false otherwise.

Name	Type	Nullable
response	boolean	No

ExecutorService.SubmitToPartition

Request Message

Type Id : 0x0905
Partition Id : the value passed in partitionId field

Name	Type	Nullable	Description
name	string	No	Name of the executor.
uuid	string	No	Unique id for the execution.
callable	byte-array	No	The callable object to be executed.
partitionId	int32	No	The id of the partition to execute this cancellation request.

Response Message

Type Id : 0x0069

The result of the callable execution.

Name	Type	Nullable
response	byte-array	Yes

ExecutorService.SubmitToAddress

Request Message

Type Id : 0x0906
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the executor.
uuid	string	No	Unique id for the execution.
callable	byte-array	No	The callable object to be executed.
address	Address	No	The member host on which the callable shall be executed on.

Response Message

Type Id : 0x0069

The result of the callable execution.

Name	Type	Nullable
response	byte-array	Yes

AtomicLong

AtomicLong.Apply

Applies a function on the value, the actual stored value will not change.

Request Message

Type Id : 0x0a01
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	The name of this IAtomicLong instance.
function	byte-array	No	The function applied to the value, the value is not changed.

Response Message

Type Id : 0x0069

The result of the function application.

Name	Type	Nullable
response	byte-array	Yes

AtomicLong.Alter

Alters the currently stored value by applying a function on it.

Request Message

Type Id : 0x0a02
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	The name of this IAtomicLong instance.
function	byte-array	No	The function applied to the currently stored value.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

AtomicLong.AlterAndGet

Alters the currently stored value by applying a function on it and gets the result.

Request Message

Type Id : 0x0a03
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	The name of this IAtomicLong instance.
function	byte-array	No	The function applied to the currently stored value.

Response Message

Type Id : 0x0067

The result of the function application.

Name	Type	Nullable
response	int64	No

AtomicLong.GetAndAlter

Alters the currently stored value by applying a function on it on and gets the old value.

Request Message

Type Id : 0x0a04
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	The name of this IAtomicLong instance.
function	byte-array	No	The function applied to the currently stored value.

Response Message

Type Id : 0x0067

The old value before the function application.

Name	Type	Nullable
response	int64	No

AtomicLong.AddAndGet

Atomically adds the given value to the current value.

Request Message

Type Id : 0x0a05
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	The name of this IAtomicLong instance.
delta	int64	No	the value to add to the current value

Response Message

Type Id : 0x0067

the updated value, the given value added to the current value

Name	Type	Nullable
response	int64	No

AtomicLong.CompareAndSet

Atomically sets the value to the given updated value only if the current value the expected value.

Request Message

Type Id : 0x0a06
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	The name of this IAtomicLong instance.
expected	int64	No	the expected value
updated	int64	No	the new value

Response Message

Type Id : 0x0065

true if successful; or false if the actual value
was not equal to the expected value.

Name	Type	Nullable
response	boolean	No

AtomicLong.DecrementAndGet

Atomically decrements the current value by one.

Request Message

Type Id : 0x0a07
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	The name of this IAtomicLong instance.

Response Message

Type Id : 0x0067

the updated value, the current value decremented by one

Name	Type	Nullable
response	int64	No

AtomicLong.Get

Gets the current value.

Request Message

Type Id : 0x0a08
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	The name of this IAtomicLong instance.

Response Message

Type Id : 0x0067

the current value

Name	Type	Nullable
response	int64	No

AtomicLong.GetAndAdd

Atomically adds the given value to the current value.

Request Message

Type Id : 0x0a09
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	The name of this IAtomicLong instance.
delta	int64	No	the value to add to the current value

Response Message

Type Id : 0x0067

the old value before the add

Name	Type	Nullable
response	int64	No

AtomicLong.GetAndSet

Atomically sets the given value and returns the old value.

Request Message

Type Id : 0x0a0a
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	The name of this IAtomicLong instance.
newValue	int64	No	the new value

Response Message

Type Id : 0x0067

the old value

Name	Type	Nullable
response	int64	No

AtomicLong.IncrementAndGet

Atomically increments the current value by one.

Request Message

Type Id : 0x0a0b
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	The name of this IAtomicLong instance.

Response Message

Type Id : 0x0067

The updated value, the current value incremented by one

Name	Type	Nullable
response	int64	No

AtomicLong.GetAndIncrement

Atomically increments the current value by one.

Request Message

Type Id : 0x0a0c
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	The name of this IAtomicLong instance.

Response Message

Type Id : 0x0067

the old value

Name	Type	Nullable
response	int64	No

AtomicLong.Set

Atomically sets the given value.

Request Message

Type Id : 0x0a0d
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	The name of this IAtomicLong instance.
newValue	int64	No	The new value

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

AtomicReference

AtomicReference.Apply

Applies a function on the value, the actual stored value will not change.

Request Message

Type Id : 0x0b01
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the AtomicReference distributed object instance.
function	byte-array	No	the function applied on the value, the stored value does not change

Response Message

Type Id : 0x0069

the result of the function application

Name	Type	Nullable
response	byte-array	Yes

AtomicReference.Alter

Alters the currently stored reference by applying a function on it.

Request Message

Type Id : 0x0b02
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the AtomicReference distributed object instance.
function	byte-array	No	the function that alters the currently stored reference

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

AtomicReference.AlterAndGet

Alters the currently stored reference by applying a function on it and gets the result.

Request Message

Type Id : 0x0b03
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the AtomicReference distributed object instance.
function	byte-array	No	the function that alters the currently stored reference

Response Message

Type Id : 0x0069

the new value, the result of the applied function.

Name	Type	Nullable
response	byte-array	Yes

AtomicReference.GetAndAlter

Alters the currently stored reference by applying a function on it on and gets the old value.

Request Message

Type Id : 0x0b04
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the AtomicReference distributed object instance.
function	byte-array	No	the function that alters the currently stored reference

Response Message

Type Id : 0x0069

the old value, the value before the function is applied

Name	Type	Nullable
response	byte-array	Yes

AtomicReference.Contains

Checks if the reference contains the value. This message is idempotent.

Request Message

Type Id : 0x0b05
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the AtomicReference distributed object instance.
expected	byte-array	Yes	the value to check (is allowed to be null).

Response Message

Type Id : 0x0065

true if the value is found, false otherwise.

Name	Type	Nullable
response	boolean	No

AtomicReference.CompareAndSet

Atomically sets the value to the given updated value only if the current value the expected value.

Request Message

Type Id : 0x0b06
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the AtomicReference distributed object instance.
expected	byte-array	Yes	the expected value
updated	byte-array	Yes	the new value

Response Message

Type Id : 0x0065

true if successful; or false if the actual value
was not equal to the expected value.

Name	Type	Nullable
response	boolean	No

AtomicReference.Get

Gets the current value. This message is idempotent.

Request Message

Type Id : 0x0b08
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the AtomicReference distributed object instance.

Response Message

Type Id : 0x0069

the current value

Name	Type	Nullable
response	byte-array	Yes

AtomicReference.Set

Atomically sets the given value.

Request Message

Type Id : 0x0b09
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the AtomicReference distributed object instance.
newValue	byte-array	Yes	the new value

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

AtomicReference.Clear

Clears the current stored reference.

Request Message

Type Id : 0x0b0a
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the AtomicReference distributed object instance.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

AtomicReference.GetAndSet

Gets the old value and sets the new value.

Request Message

Type Id : 0x0b0b
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the AtomicReference distributed object instance.
newValue	byte-array	Yes	the new value.

Response Message

Type Id : 0x0069

the old value.

Name	Type	Nullable
response	byte-array	Yes

AtomicReference.SetAndGet

Sets and gets the value.

Request Message

Type Id : 0x0b0c
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the AtomicReference distributed object instance.
newValue	byte-array	Yes	the new value

Response Message

Type Id : 0x0069

the new value

Name	Type	Nullable
response	byte-array	Yes

AtomicReference.IsNull

Checks if the stored reference is null. This message is idempotent.

Request Message

Type Id : 0x0b0d
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the AtomicReference distributed object instance.

Response Message

Type Id : 0x0065

true if null, false otherwise.

Name	Type	Nullable
response	boolean	No

CountdownLatch

CountdownLatch.Await

Causes the current thread to wait until the latch has counted down to zero, or an exception is thrown, or the specified waiting time elapses. If the current count is zero then this method returns immediately with the value true. If the current count is greater than zero, then the current thread becomes disabled for thread scheduling purposes and lies dormant until one of five things happen: the count reaches zero due to invocations of the #countDown method, this ICountDownLatch instance is destroyed, the countdown owner becomes disconnected, some other thread Thread#interrupt interrupts the current thread, or the specified waiting time elapses. If the count reaches zero, then the method returns with the value true. If the current thread: has its interrupted status set on entry to this method, or is Thread#interrupt interrupted while waiting, then INTERRUPTED is thrown and the current thread's interrupted status is cleared. If the specified waiting time elapses then the value false is returned. If the time is less than or equal to zero, the method will not wait at all.

Request Message

Type Id : 0x0c01
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the CountDownLatch
timeout	int64	No	The maximum time in milliseconds to wait

Response Message

Type Id : 0x0065

True if the count reached zero, false if the waiting time elapsed before the count reached zero

Name	Type	Nullable
response	boolean	No

CountdownLatch.CountDown

Decrements the count of the latch, releasing all waiting threads if the count reaches zero. If the current count is greater than zero, then it is decremented. If the new count is zero: All waiting threads are re-enabled for thread scheduling purposes, and Countdown owner is set to null. If the current count equals zero, then nothing happens.

Request Message

Type Id : 0x0c02
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the CountDownLatch

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

CountdownLatch.GetCount

Returns the current count. This message is idempotent.

Request Message

Type Id : 0x0c03
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the CountDownLatch

Response Message

Type Id : 0x0066

The current count for the latch.

Name	Type	Nullable
response	int32	No

CountdownLatch.TrySetCount

Sets the count to the given value if the current count is zero. If the count is not zero, then this method does nothing and returns false

Request Message

Type Id : 0x0c04
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the CountDownLatch
count	int32	No	The number of times countDown must be invoked before threads can pass through await

Response Message

Type Id : 0x0065

True if the new count was set, false if the current count is not zero.

Name	Type	Nullable
response	boolean	No

Semaphore

Semaphore.Init

Try to initialize this ISemaphore instance with the given permit count

Request Message

Type Id : 0x0d01
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Semaphore
permits	int32	No	The given permit count

Response Message

Type Id : 0x0065

True if initialization succeeds, false otherwise.

Name	Type	Nullable
response	boolean	No

Semaphore.Acquire

Acquires the given number of permits if they are available, and returns immediately, reducing the number of available permits by the given amount. If insufficient permits are available then the current thread becomes disabled for thread scheduling purposes and lies dormant until one of three things happens: some other thread invokes one of the methods for this semaphore, the current thread is next to be assigned permits and the number of available permits satisfies this request, this ISemaphore instance is destroyed, or some other thread the current thread. If the current thread has its interrupted status set on entry to this method, or is while waiting for a permit, then is thrown and the current thread's interrupted status is cleared.

Request Message

Type Id : 0x0d02
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Semaphore
permits	int32	No	The given permit count

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Semaphore.AvailablePermits

Returns the current number of permits currently available in this semaphore. This method is typically used for debugging and testing purposes. This message is idempotent.

Request Message

Type Id : 0x0d03
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Semaphore

Response Message

Type Id : 0x0066

The number of permits available in this semaphore.

Name	Type	Nullable
response	int32	No

Semaphore.DrainPermits

Acquires and returns all permits that are immediately available.

Request Message

Type Id : 0x0d04
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Semaphore

Response Message

Type Id : 0x0066

The number of permits drained

Name	Type	Nullable
response	int32	No

Semaphore.ReducePermits

Shrinks the number of available permits by the indicated reduction. This method differs from acquire in that it does not block waiting for permits to become available.

Request Message

Type Id : 0x0d05
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Semaphore
reduction	int32	No	The number of permits to remove

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Semaphore.Release

Releases the given number of permits, increasing the number of available permits by that amount. There is no requirement that a thread that releases a permit must have acquired that permit by calling one of the acquire()acquire methods. Correct usage of a semaphore is established by programming convention in the application.

Request Message

Type Id : 0x0d06
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Semaphore
permits	int32	No	The number of permits to remove

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Semaphore.TryAcquire

Acquires the given number of permits, if they are available, and returns immediately, with the value true, reducing the number of available permits by the given amount. If insufficient permits are available then this method will return immediately with the value false and the number of available permits is unchanged.

Request Message

Type Id : 0x0d07
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Semaphore
permits	int32	No	The number of permits to remove
timeout	int64	No	The maximum time to wait for a permit

Response Message

Type Id : 0x0065

true if all permits were acquired, false if the waiting time elapsed before all permits could be acquired

Name	Type	Nullable
response	boolean	No

ReplicatedMap

ReplicatedMap.Put

Associates a given value to the specified key and replicates it to the cluster. If there is an old value, it will be replaced by the specified one and returned from the call. In addition, you have to specify a ttl and its TimeUnit to define when the value is outdated and thus should be removed from the replicated map.

Request Message

Type Id : 0x0e01
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the ReplicatedMap
key	byte-array	No	Key with which the specified value is to be associated.
value	byte-array	No	Value to be associated with the specified key
ttl	int64	No	ttl in milliseconds to be associated with the specified key-value pair

Response Message

Type Id : 0x0069

The old value if existed for the key.

Name	Type	Nullable
response	byte-array	Yes

ReplicatedMap.Size

Returns the number of key-value mappings in this map. If the map contains more than Integer.MAX_VALUE elements, returns Integer.MAX_VALUE. This message is idempotent.

Request Message

Type Id : 0x0e02
Partition Id : a random partition id from 0 to PARTITION_COUNT(PARTITION_COUNT exclusive)

Name	Type	Nullable	Description
name	string	No	Name of the ReplicatedMap

Response Message

Type Id : 0x0066

the number of key-value mappings in this map.

Name	Type	Nullable
response	int32	No

ReplicatedMap.IsEmpty

Return true if this map contains no key-value mappings This message is idempotent.

Request Message

Type Id : 0x0e03
Partition Id : a random partition id from 0 to PARTITION_COUNT(PARTITION_COUNT exclusive)

Name	Type	Nullable	Description
name	string	No	Name of the ReplicatedMap

Response Message

Type Id : 0x0065

True if this map contains no key-value mappings

Name	Type	Nullable
response	boolean	No

ReplicatedMap.ContainsKey

Returns true if this map contains a mapping for the specified key. This message is idempotent.

Request Message

Type Id : 0x0e04
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the ReplicatedMap
key	byte-array	No	The key whose associated value is to be returned.

Response Message

Type Id : 0x0065

True if this map contains a mapping for the specified key

Name	Type	Nullable
response	boolean	No

ReplicatedMap.ContainsValue

Returns true if this map maps one or more keys to the specified value.
This operation will probably require time linear in the map size for most implementations of the Map interface. This message is idempotent.

Request Message

Type Id : 0x0e05
Partition Id : a random partition id from 0 to PARTITION_COUNT(PARTITION_COUNT exclusive)

Name	Type	Nullable	Description
name	string	No	Name of the ReplicatedMap
value	byte-array	No	value whose presence in this map is to be tested

Response Message

Type Id : 0x0065

true if this map maps one or more keys to the specified value

Name	Type	Nullable
response	boolean	No

ReplicatedMap.Get

Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key.
If this map permits null values, then a return value of null does not
necessarily indicate that the map contains no mapping for the key; it's also possible that the map
explicitly maps the key to null. The #containsKey operation may be used to distinguish these two cases. This message is idempotent.

Request Message

Type Id : 0x0e06
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the ReplicatedMap
key	byte-array	No	The key whose associated value is to be returned

Response Message

Type Id : 0x0069

The value to which the specified key is mapped, or null if this map contains no mapping for the key

Name	Type	Nullable
response	byte-array	Yes

ReplicatedMap.Remove

Removes the mapping for a key from this map if it is present (optional operation). Returns the value to which this map previously associated the key,
or null if the map contained no mapping for the key. If this map permits null values, then a return value of
null does not necessarily indicate that the map contained no mapping for the key; it's also possible that the map
explicitly mapped the key to null. The map will not contain a mapping for the specified key once the call returns.

Request Message

Type Id : 0x0e07
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the ReplicatedMap
key	byte-array	No	Key with which the specified value is to be associated.

Response Message

Type Id : 0x0069

the previous value associated with key, or null if there was no mapping for key.

Name	Type	Nullable
response	byte-array	Yes

ReplicatedMap.PutAll

Copies all of the mappings from the specified map to this map (optional operation). The effect of this call is equivalent to that of calling put(Object,Object) put(k, v) on this map once for each mapping from key k to value v in the specified map. The behavior of this operation is undefined if the specified map is modified while the operation is in progress.

Request Message

Type Id : 0x0e08
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the ReplicatedMap
entries	array of key-value byte array pair	No	entries to be stored in this map

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

ReplicatedMap.Clear

The clear operation wipes data out of the replicated maps.It is the only synchronous remote operation in this implementation, so be aware that this might be a slow operation. If some node fails on executing the operation, it is retried for at most 3 times (on the failing nodes only). If it does not work after the third time, this method throws a OPERATION_TIMEOUT back to the caller.

Request Message

Type Id : 0x0e09
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Replicated Map

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

ReplicatedMap.AddEntryListenerToKeyWithPredicate

Adds an continuous entry listener for this map. The listener will be notified for map add/remove/update/evict

events filtered by the given predicate.

Request Message

Type Id : 0x0e0a
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Replicated Map
key	byte-array	No	Key with which the specified value is to be associated.
predicate	byte-array	No	The predicate for filtering entries
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

A unique string which is used as a key to remove the listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cb

Name	Type	Nullable	Description
key	byte-array	Yes	The key for the entry in the map.
value	byte-array	Yes	The value of the entry in the map.
oldValue	byte-array	Yes	The original value for the key in the map if exists.
mergingValue	byte-array	Yes	The incoming merging value of the entry event.
eventType	int32	No	Possible types are: ADDED(1) REMOVED(2) UPDATED(3) EVICTED(4) EVICT_ALL(5) CLEAR_ALL(6) MERGED(7)
uuid	string	No	The id of the member.
numberOfAffectedEntries	int32	No	The number of entries affected in the map.

ReplicatedMap.AddEntryListenerWithPredicate

Adds an continuous entry listener for this map. The listener will be notified for map add/remove/update/evict events filtered by the given predicate.

Request Message

Type Id : 0x0e0b
Partition Id : -1

Name	Type	Nullable	Description

name	string	No	Name of the Replicated Map
predicate	byte-array	No	The predicate for filtering entries
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

A unique string which is used as a key to remove the listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cb

Name	Type	Nullable	Description
key	byte-array	Yes	The key for the entry in the map.
value	byte-array	Yes	The value of the entry in the map.
oldValue	byte-array	Yes	The original value for the key in the map if exists.
mergingValue	byte-array	Yes	The incoming merging value of the entry event.
eventType	int32	No	Possible types are: ADDED(1) REMOVED(2) UPDATED(3) EVICTED(4) EVICT_ALL(5) CLEAR_ALL(6) MERGED(7)
uuid	string	No	The id of the member.
numberOfAffectedEntries	int32	No	The number of entries affected in the map.

ReplicatedMap.AddEntryListenerToKey

Adds the specified entry listener for the specified key. The listener will be notified for all add/remove/update/evict events of the specified key only.

Request Message

Type Id : 0x0e0c
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Replicated Map
key	byte-array	No	Key with which the specified value is to be associated.
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

A unique string which is used as a key to remove the listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cb

Name	Type	Nullable	Description
key	byte-array	Yes	The key for the entry in the map.
value	byte-array	Yes	The value of the entry in the map.
oldValue	byte-array	Yes	The original value for the key in the map if exists.
mergingValue	byte-array	Yes	The incoming merging value of the entry event.
eventType	int32	No	Possible types are: ADDED(1) REMOVED(2) UPDATED(3) EVICTED(4) EVICT_ALL(5) CLEAR_ALL(6) MERGED(7)
uuid	string	No	The id of the member.
numberOfAffectedEntries	int32	No	The number of entries affected in the map.

ReplicatedMap.AddEntryListener

Adds an entry listener for this map. The listener will be notified for all map add/remove/update/evict events.

Request Message

Type Id : 0x0e0d
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the ReplicatedMap
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

A unique string which is used as a key to remove the listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cb

Name	Type	Nullable	Description
key	byte-array	Yes	The key for the entry in the map.
value	byte-array	Yes	The value of the entry in the map.

oldValue	byte-array	Yes	The original value for the key in the map if exists.
mergingValue	byte-array	Yes	The incoming merging value of the entry event.
eventType	int32	No	Possible types are: ADDED(1) REMOVED(2) UPDATED(3) EVICTED(4) EVICT_ALL(5) CLEAR_ALL(6) MERGED(7)
uuid	string	No	The id of the member.
numberOfAffectedEntries	int32	No	The number of entries affected in the map.

ReplicatedMap.RemoveEntryListener

Removes the specified entry listener. Returns silently if there was no such listener added before. This message is idempotent.

Request Message

Type Id : 0x0e0e
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the ReplicatedMap
registrationId	string	No	ID of the registered entry listener.

Response Message

Type Id : 0x0065

True if registration is removed, false otherwise.

Name	Type	Nullable
response	boolean	No

ReplicatedMap.KeySet

Returns a lazy Set view of the key contained in this map. A LazySet is optimized for querying speed (preventing eager deserialization and hashing on HashSet insertion) and does NOT provide all operations. Any kind of mutating function will throw an UNSUPPORTED_OPERATION. Same is true for operations like java.util.Set#contains(Object) and java.util.Set#containsAll(java.util.Collection) which would result in very poor performance if called repeatedly (for example, in a loop). If the use case is different from querying the data, please copy the resulting set into a new java.util.HashSet. This message is idempotent.

Request Message

Type Id : 0x0e0f
Partition Id : a random partition id from 0 to PARTITION_COUNT(PARTITION_COUNT exclusive)

Name	Type	Nullable	Description
name	string	No	Name of the ReplicatedMap

Response Message

Type Id : 0x006a

A lazy set view of the keys contained in this map.

Name	Type	Nullable
response	array of byte-array	No

ReplicatedMap.Values

This message is idempotent.

Request Message

Type Id : 0x0e10
Partition Id : a random partition id from 0 to PARTITION_COUNT(PARTITION_COUNT exclusive)

Name	Type	Nullable	Description
name	string	No	Name of the ReplicatedMap

Response Message

Type Id : 0x006a

A collection view of the values contained in this map.

Name	Type	Nullable
response	array of byte-array	No

ReplicatedMap.EntrySet

This message is idempotent.

Request Message

Type Id : 0x0e11
Partition Id : a random partition id from 0 to PARTITION_COUNT(PARTITION_COUNT exclusive)

Name	Type	Nullable	Description
name	string	No	Name of the ReplicatedMap

Response Message

Type Id : 0x0075

A lazy set view of the mappings contained in this map.

Name	Type	Nullable
response	array of key-value byte array pair	No

ReplicatedMap.AddNearCacheEntryListener

Request Message

Type Id : 0x0e12
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the ReplicatedMap
includeValue	boolean	No	True if EntryEvent should contain the value,false otherwise
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

A unique string which is used as a key to remove the listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00cb

Name	Type	Nullable	Description
key	byte-array	Yes	The key for the entry in the map.
value	byte-array	Yes	The value of the entry in the map.
oldValue	byte-array	Yes	The original value for the key in the map if exists.
mergingValue	byte-array	Yes	The incoming merging value of the entry event.
eventType	int32	No	Possible types are: ADDED(1) REMOVED(2) UPDATED(3) EVICTED(4) EVICT_ALL(5) CLEAR_ALL(6) MERGED(7)
uuid	string	No	The id of the member.
numberOfAffectedEntries	int32	No	The number of entries affected in the map.

MapReduce

MapReduce.Cancel

Request Message

Type Id : 0x0f01
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the distributed object
jobId	string	No	Id of the job to cancel

Response Message

Type Id : 0x0065

Returns true if successful, false otherwise

Name	Type	Nullable
response	boolean	No

MapReduce.JobProcessInformation

This message is idempotent.

Request Message

Type Id : 0x0f02
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the distributed object
jobId	string	No	Id of the job

Response Message

Type Id : 0x0070

The information about the job if exists

Name	Type	Nullable
jobPartitionStates	array of Job Partition State	No
processRecords	int32	No

MapReduce.ForMap

Request Message

Type Id : 0x0f03
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the distributed object
jobId	string	No	Id of the job
predicate	byte-array	Yes	The filter to use during operation
mapper	byte-array	No	The mapper for the operation
combinerFactory	byte-array	Yes	The combiner factory to use

reducerFactory	byte-array	Yes	The reducer factory to be used
mapName	string	No	Name of the Map object to work on.
chunkSize	int32	No	The number of items for which the reduce shall be performed
keys	array of byte-array	Yes	The keys for the objects to be processed
topologyChangedStrategy	string	Yes	The strategy to use if a topology change is detected.

Response Message

Type Id : 0x0075

The resulting key-value pairs.

Name	Type	Nullable
response	array of key-value byte array pair	No

MapReduce.ForList

Request Message

Type Id : 0x0f04
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the distributed object
jobId	string	No	Id of the job
predicate	byte-array	Yes	The filter to use during operation
mapper	byte-array	No	The mapper for the operation
combinerFactory	byte-array	Yes	The combiner factory to use
reducerFactory	byte-array	Yes	The reducer factory to be used
listName	string	No	Name of the List object to work on.
chunkSize	int32	No	The number of items for which the reduce shall be performed
keys	array of byte-array	Yes	The keys for the objects to be processed
topologyChangedStrategy	string	Yes	The strategy to use if a topology change is detected.

Response Message

Type Id : 0x0075

The resulting key-value pairs.

Name	Type	Nullable
response	array of key-value byte array pair	No

MapReduce.ForSet

Request Message

Type Id : 0x0f05

Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the distributed object
jobId	string	No	Id of the job
predicate	byte-array	Yes	The filter to use during operation
mapper	byte-array	No	The mapper for the operation
combinerFactory	byte-array	Yes	The combiner factory to use
reducerFactory	byte-array	Yes	The reducer factory to be used
setName	string	No	Name of the Set object to work on.
chunkSize	int32	No	The number of items for which the reduce shall be performed
keys	array of byte-array	Yes	The keys for the objects to be processed
topologyChangedStrategy	string	Yes	The strategy to use if a topology change is detected.

Response Message

Type Id : 0x0075

The resulting key-value pairs.

Name	Type	Nullable
response	array of key-value byte array pair	No

MapReduce.ForMultiMap

Request Message

Type Id : 0x0f06
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the distributed object
jobId	string	No	Id of the job
predicate	byte-array	Yes	The filter to use during operation
mapper	byte-array	No	The mapper for the operation
combinerFactory	byte-array	Yes	The combiner factory to use
reducerFactory	byte-array	Yes	The reducer factory to be used
multiMapName	string	No	Name of the MultiMap object to work on.
chunkSize	int32	No	The number of items for which the reduce shall be performed
keys	array of byte-array	Yes	The keys for the objects to be processed
topologyChangedStrategy	string	Yes	The strategy to use if a topology change is detected.

Response Message

Type Id : 0x0075

The resulting key-value pairs.

Name	Type	Nullable

response	array of key-value byte array pair	No

MapReduce.ForCustom

Request Message

Type Id : 0x0f07
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the distributed object
jobId	string	No	Id of the job
predicate	byte-array	Yes	The filter to use during operation
mapper	byte-array	No	The mapper for the operation
combinerFactory	byte-array	Yes	The combiner factory to use
reducerFactory	byte-array	Yes	The reducer factory to be used
keyValueSource	byte-array	No	custom data sources for mapreduce algorithm. The object implements the com.hazelcast.mapreduce.KeyValueSource interface
chunkSize	int32	No	The number of items for which the reduce shall be performed
keys	array of byte-array	Yes	The keys for the objects to be processed
topologyChangedStrategy	string	Yes	The strategy to use if a topology change is detected.

Response Message

Type Id : 0x0075

The resulting key-value pairs.

Name	Type	Nullable
response	array of key-value byte array pair	No

TransactionalMap

TransactionalMap.ContainsKey

Returns true if this map contains an entry for the specified key.

Request Message

Type Id : 0x1001
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map

---

txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	The specified key.

Response Message

Type Id : 0x0065

True if this map contains an entry for the specified key.

Name	Type	Nullable
response	boolean	No

TransactionalMap.Get

Returns the value for the specified key, or null if this map does not contain this key.

Request Message

Type Id : 0x1002
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	The specified key

Response Message

Type Id : 0x0069

The value for the specified key

Name	Type	Nullable
response	byte-array	Yes

TransactionalMap.GetForUpdate

Locks the key and then gets and returns the value to which the specified key is mapped. Lock will be released at the end of the transaction (either commit or rollback).

Request Message

Type Id : 0x1003
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map

txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	The value to which the specified key is mapped

Response Message

Type Id : 0x0069

The value for the specified key

Name	Type	Nullable
response	byte-array	Yes

TransactionalMap.Size

Returns the number of entries in this map.

Request Message

Type Id : 0x1004
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0066

The number of entries in this map.

Name	Type	Nullable
response	int32	No

TransactionalMap.IsEmpty

Returns true if this map contains no entries.

Request Message

Type Id : 0x1005
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map
txnId	string	No	ID of the this transaction operation

threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0065

true if this map contains no entries.

Name	Type	Nullable
response	boolean	No

TransactionalMap.Put

Associates the specified value with the specified key in this map. If the map previously contained a mapping for the key, the old value is replaced by the specified value. The object to be put will be accessible only in the current transaction context till transaction is committed.

Request Message

Type Id : 0x1006
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	The specified key
value	byte-array	No	The value to associate with the key.
ttl	int64	No	The duration in milliseconds after which this entry shall be deleted. O means infinite.

Response Message

Type Id : 0x0069

Previous value associated with key or null if there was no mapping for key

Name	Type	Nullable
response	byte-array	Yes

TransactionalMap.Set

Associates the specified value with the specified key in this map. If the map previously contained a mapping for the key, the old value is replaced by the specified value. This method is preferred to #put(Object, Object) if the old value is not needed.
The object to be set will be accessible only in the current transaction context until the transaction is committed.

Request Message

Type Id : 0x1007
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	The specified key
value	byte-array	No	The value to associate with key

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

TransactionalMap.PutIfAbsent

If the specified key is not already associated with a value, associate it with the given value.
The object to be put will be accessible only in the current transaction context until the transaction is committed.

Request Message

Type Id : 0x1008
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	The specified key
value	byte-array	No	The value to associate with the key when there is no previous value.

Response Message

Type Id : 0x0069

The previous value associated with key, or null if there was no mapping for key.

Name	Type	Nullable
response	byte-array	Yes

TransactionalMap.Replace

Replaces the entry for a key only if it is currently mapped to some value. The object to be replaced will be accessible only in the current transaction context until the transaction is committed.

Request Message

Type Id : 0x1009
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	The specified key
value	byte-array	No	The value replaced the previous value

Response Message

Type Id : 0x0069

The previous value associated with key, or null if there was no mapping for key.

Name	Type	Nullable
response	byte-array	Yes

TransactionalMap.ReplaceIfSame

Replaces the entry for a key only if currently mapped to a given value. The object to be replaced will be accessible only in the current transaction context until the transaction is committed.

Request Message

Type Id : 0x100a
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	The specified key.
oldValue	byte-array	No	Replace the key value if it is the old value.
newValue	byte-array	No	The new value to replace the old value.

Response Message

Type Id : 0x0065

true if the value was replaced.

Name	Type	Nullable
response	boolean	No

TransactionalMap.Remove

Removes the mapping for a key from this map if it is present. The map will not contain a mapping for the specified key once the call returns. The object to be removed will be accessible only in the current transaction context until the transaction is committed.

Request Message

Type Id : 0x100b
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	Remove the mapping for this key.

Response Message

Type Id : 0x0069

The previous value associated with key, or null if there was no mapping for key

Name	Type	Nullable
response	byte-array	Yes

TransactionalMap.Delete

Removes the mapping for a key from this map if it is present. The map will not contain a mapping for the specified key once the call returns. This method is preferred to #remove(Object) if the old value is not needed. The object to be deleted will be removed from only the current transaction context until the transaction is committed.

Request Message

Type Id : 0x100c
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	Remove the mapping for this key.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

TransactionalMap.RemoveIfSame

Removes the entry for a key only if currently mapped to a given value. The object to be removed will be removed from only the current transaction context until the transaction is committed.

Request Message

Type Id : 0x100d
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	The specified key
value	byte-array	No	Remove the key if it has this value.

Response Message

Type Id : 0x0065

True if the value was removed

Name	Type	Nullable
response	boolean	No

TransactionalMap.KeySet

Returns a set clone of the keys contained in this map. The set is NOT backed by the map, so changes to the map are NOT reflected in the set, and vice-versa. This method is always executed by a distributed query, so it may throw a QueryResultSizeExceededException if query result size limit is configured.

Request Message

Type Id : 0x100e
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x006a

A set clone of the keys contained in this map.

Name	Type	Nullable
response	array of byte-array	No

TransactionalMap.KeySetWithPredicate

Queries the map based on the specified predicate and returns the keys of matching entries. Specified predicate runs on all members in parallel.The set is NOT backed by the map, so changes to the map are NOT reflected in the set, and vice-versa. This method is always executed by a distributed query, so it may throw a QueryResultSizeExceededException if query result size limit is configured.

Request Message

Type Id : 0x100f
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
predicate	byte-array	No	Specified query criteria.

Response Message

Type Id : 0x006a

Result key set for the query.

Name	Type	Nullable
response	array of byte-array	No

TransactionalMap.Values

Returns a collection clone of the values contained in this map. The collection is NOT backed by the map,
so changes to the map are NOT reflected in the collection, and vice-versa. This method is always executed by a
distributed query, so it may throw a QueryResultSizeExceededException if query result size limit is configured.

Request Message

Type Id : 0x1010
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x006a

All values in the map

Name	Type	Nullable
response	array of byte-array	No

TransactionalMap.ValuesWithPredicate

Queries the map based on the specified predicate and returns the values of matching entries.Specified predicate
runs on all members in parallel. The collection is NOT backed by the map, so changes to the map are NOT reflected
in the collection, and vice-versa. This method is always executed by a distributed query, so it may throw

a QueryResultSizeExceededException if query result size limit is configured.

Request Message

Type Id : 0x1011
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
predicate	byte-array	No	Specified query criteria.

Response Message

Type Id : 0x006a

Result value collection of the query.

Name	Type	Nullable
response	array of byte-array	No

TransactionalMultiMap

TransactionalMultiMap.Put

Stores a key-value pair in the multimap.

Request Message

Type Id : 0x1101
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Multi Map
txnId	string	No	ID of the transaction
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	The key to be stored
value	byte-array	No	The value to be stored

Response Message

Type Id : 0x0065

True if the size of the multimap is increased, false if the multimap already contains the key-value pair.

Name	Type	Nullable
response	boolean	No

TransactionalMultiMap.Get

Returns the collection of values associated with the key.

Request Message

Type Id : 0x1102
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Multi Map
txnId	string	No	ID of the transaction
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	The key whose associated values are returned

Response Message

Type Id : 0x006a

The collection of the values associated with the key

Name	Type	Nullable
response	array of byte-array	No

TransactionalMultiMap.Remove

Removes the given key value pair from the multimap.

Request Message

Type Id : 0x1103
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Multi Map
txnId	string	No	ID of the transaction
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	The key whose associated values are returned

Response Message

Type Id : 0x006a

True if the size of the multimap changed after the remove operation, false otherwise.

Name	Type	Nullable
response	array of byte-array	No

TransactionalMultiMap.RemoveEntry

Removes all the entries associated with the given key.

Request Message

Type Id : 0x1104
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Multi Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	The key whose associated values are returned
value	byte-array	No	The value to be stored

Response Message

Type Id : 0x0065

True if the size of the multimap changed after the remove operation, false otherwise.

Name	Type	Nullable
response	boolean	No

TransactionalMultiMap.ValueCount

Returns the number of values matching the given key in the multimap.

Request Message

Type Id : 0x1105
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Multi Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
key	byte-array	No	The key whose number of values are returned

Response Message

Type Id : 0x0066

The number of values matching the given key in the multimap

Name	Type	Nullable
response	int32	No

TransactionalMultiMap.Size

Returns the number of key-value pairs in the multimap.

Request Message

Type Id : 0x1106
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Multi Map
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0066

The number of key-value pairs in the multimap

Name	Type	Nullable
response	int32	No

TransactionalSet

TransactionalSet.Add

Add new item to transactional set.

Request Message

Type Id : 0x1201
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Set
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
item	byte-array	No	Item added to transactional set

Response Message

Type Id : 0x0065

True if item is added successfully

Name	Type	Nullable
response	boolean	No

TransactionalSet.Remove

Remove item from transactional set.

Request Message

Type Id : 0x1202
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Set
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
item	byte-array	No	Item removed from Transactional Set

Response Message

Type Id : 0x0065

True if item is remove successfully

Name	Type	Nullable
response	boolean	No

TransactionalSet.Size

Returns the size of the set.

Request Message

Type Id : 0x1203
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Set
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0066

The size of the set

Name	Type	Nullable
response	int32	No

TransactionalList

TransactionalList.Add

Adds a new item to the transactional list.

Request Message

Type Id : 0x1301
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional List
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
item	byte-array	No	The new item added to the transactionalList

Response Message

Type Id : 0x0065

True if the item is added successfully, false otherwise

Name	Type	Nullable
response	boolean	No

TransactionalList.Remove

Remove item from the transactional list

Request Message

Type Id : 0x1302
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional List
txnId	string	No	ID of the this transaction operation

---

threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
item	byte-array	No	Item to remove to transactional List

Response Message

Type Id : 0x0065

True if the removed successfully,false otherwise

Name	Type	Nullable
response	boolean	No

TransactionalList.Size

Returns the size of the list

Request Message

Type Id : 0x1303
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional List
txnId	string	No	ID of the this transaction operation
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0066

The size of the list

Name	Type	Nullable
response	int32	No

TransactionalQueue

TransactionalQueue.Offer

Inserts the specified element into this queue, waiting up to the specified wait time if necessary for space to become available.

Request Message

Type Id : 0x1401

Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transcational Queue
txnId	string	No	ID of the transaction
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
item	byte-array	No	The element to add
timeout	int64	No	How long to wait before giving up, in milliseconds

Response Message

Type Id : 0x0065

true if successful, or false if the specified waiting time elapses before space is available

Name	Type	Nullable
response	boolean	No

TransactionalQueue.Take

Retrieves and removes the head of this queue, waiting if necessary until an element becomes available.

Request Message

Type Id : 0x1402
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Queue
txnId	string	No	ID of the transaction
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0069

The head of this queue

Name	Type	Nullable
response	byte-array	Yes

TransactionalQueue.Poll

Retrieves and removes the head of this queue, waiting up to the specified wait time if necessary for an element to become available.

Request Message

Type Id : 0x1403

Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Queue
txnId	string	No	ID of the transaction
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
timeout	int64	No	How long to wait before giving up, in milliseconds

Response Message

Type Id : 0x0069

The head of this queue, or null if the specified waiting time elapses before an element is available

Name	Type	Nullable
response	byte-array	Yes

TransactionalQueue.Peek

Retrieves, but does not remove, the head of this queue, or returns null if this queue is empty.

Request Message

Type Id : 0x1404
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Queue
txnId	string	No	ID of the transaction
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.
timeout	int64	No	How long to wait before giving up, in milliseconds

Response Message

Type Id : 0x0069

The value at the head of the queue.

Name	Type	Nullable
response	byte-array	Yes

TransactionalQueue.Size

Returns the number of elements in this collection.If this collection contains more than Integer.MAX_VALUE elements, returns Integer.MAX_VALUE.

Request Message

Type Id : 0x1405

Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Transactional Queue
txnId	string	No	ID of the transaction
threadId	int64	No	The id of the user thread performing the operation. It is used to guarantee that only the lock holder thread (if a lock exists on the entry) can perform the requested operation.

Response Message

Type Id : 0x0066

The number of elements in this collection

Name	Type	Nullable
response	int32	No

Cache

Cache.AddEntryListener

Request Message

Type Id : 0x1501
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the cache.
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

Registration id for the registered listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00d2

Name	Type	Nullable	Description

---

type	int32	No	The type of the event. Possible values for the event are: CREATED(1): An event type indicating that the cache entry was created. UPDATED(2): An event type indicating that the cache entry was updated, i.e. a previous mapping existed. REMOVED(3): An event type indicating that the cache entry was removed. EXPIRED(4): An event type indicating that the cache entry has expired. EVICTED(5): An event type indicating that the cache entry has evicted. INVALIDATED(6): An event type indicating that the cache entry has invalidated for near cache invalidation. COMPLETED(7): An event type indicating that the cache operation has completed. EXPIRATION_TIME_UPDATED(8): An event type indicating that the expiration time of cache record has been updated
keys	array of Cache Event Data	No	The keys for the entries in the cache.
completionId	int32	No	User generated id which shall be received as a field of the cache event upon completion of the request in the cluster.

Cache.AddInvalidationListener

Request Message

Type Id : 0x1502
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the cache.
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

Registration id for the registered listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00d0

Name	Type	Nullable	Description
name	string	No	Name of the cache.
key	byte-array	Yes	The key for the entry.
sourceUuid	string	Yes	The id of the server member.

Event Message

Type Id : 0x00d3

Name	Type	Nullable	Description
name	string	No	Name of the cache.
keys	array of byte-array	No	The keys for the entries in batch invalidation.
sourceUuids	array of string	Yes	The ids of the server members.

Cache.Clear

Clears the contents of the cache, without notifying listeners or CacheWriters.

Request Message

Type Id : 0x1503
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the cache.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Cache.RemoveAllKeys

Removes entries for the specified keys. The order in which the individual entries are removed is undefined.
For every entry in the key set, the following are called: any registered CacheEntryRemovedListeners if the cache is a write-through cache, the CacheWriter. If the key set is empty, the CacheWriter is not called.

Request Message

Type Id : 0x1504
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the cache.
keys	array of byte-array	No	The keys to remove.
completionId	int32	No	User generated id which shall be received as a field of the cache event upon completion of the request in the cluster.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Cache.RemoveAll

Removes all of the mappings from this cache. The order that the individual entries are removed is undefined.
For every mapping that exists the following are called: any registered CacheEntryRemovedListener if the cache is a write-through cache, the CacheWriter.If the cache is empty, the CacheWriter is not called.
This is potentially an expensive operation as listeners are invoked. Use #clear() to avoid this.

Request Message

Type Id : 0x1505
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the cache.
completionId	int32	No	User generated id which shall be received as a field of the cache event upon completion of the request in the cluster.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Cache.ContainsKey

Determines if the Cache contains an entry for the specified key. More formally, returns true if and only if this
cache contains a mapping for a key k such that key.equals(k). (There can be at most one such mapping.) This message is idempotent.

Request Message

Type Id : 0x1506
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the cache.
key	byte-array	No	The key whose presence in this cache is to be tested.

Response Message

Type Id : 0x0065

Returns true if cache value for the key exists, false otherwise.

Name	Type	Nullable
response	boolean	No

Cache.CreateConfig

This message is idempotent.

Request Message

Type Id : 0x1507
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
cacheConfig	byte-array	No	The cache configuration. Byte-array which is serialized from an object implementing javax.cache.configuration.Configuration interface.
createAlsoOnOthers	boolean	No	True if the configuration shall be created on all members, false otherwise.

Response Message

Type Id : 0x0069

The created configuration object. Byte-array which is serialized from an object implementing javax.cache.configuration.Configuration interface.

Name	Type	Nullable
response	byte-array	Yes

Cache.Destroy

Closes the cache. Clears the internal content and releases any resource.

Request Message

Type Id : 0x1508
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the cache.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Cache.EntryProcessor

Request Message

Type Id : 0x1509
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the cache.
key	byte-array	No	the key to the entry
entryProcessor	byte-array	No	Entry processor to invoke. Byte-array which is serialized from an object implementing javax.cache.processor.EntryProcessor.
arguments	array of byte-array	No	additional arguments to pass to the EntryProcessor
completionId	int32	No	User generated id which shall be received as a field of the cache event upon completion of the request in the cluster.

Response Message

Type Id : 0x0069

the result of the processing, if any, defined by the EntryProcessor implementation

Name	Type	Nullable
response	byte-array	Yes

Cache.GetAll

Gets a collection of entries from the cache with custom expiry policy, returning them as Map of the values associated with the set of keys requested. If the cache is configured for read-through operation mode, the underlying configured javax.cache.integration.CacheLoader might be called to retrieve the values of the keys from any kind of external resource.

Request Message

Type Id : 0x150a
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the cache.
keys	array of byte-array	No	The keys whose associated values are to be returned.
expiryPolicy	byte-array	Yes	Expiry policy for the entry. Byte-array which is serialized from an object implementing javax.cache.expiry.ExpiryPolicy interface.

Response Message

Type Id : 0x0075

A map of entries that were found for the given keys. Keys not found in the cache are not in the returned map.

Name	Type	Nullable
response	array of key-value byte array pair	No

Cache.GetAndRemove

Atomically removes the entry for a key only if currently mapped to some value.

Request Message

Type Id : 0x150b
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the cache.
key	byte-array	No	key with which the specified value is associated
completionId	int32	No	User generated id which shall be received as a field of the cache event upon completion of the request in the cluster.

Response Message

Type Id : 0x0069

the value if one existed or null if no mapping existed for this key

Name	Type	Nullable
response	byte-array	Yes

Cache.GetAndReplace

Atomically replaces the assigned value of the given key by the specified value using a custom javax.cache.expiry.ExpiryPolicy and returns the previously assigned value. If the cache is configured for write-through operation mode, the underlying configured javax.cache.integration.CacheWriter might be called to store the value of the key to any kind of external resource.

Request Message

Type Id : 0x150c
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the cache.
key	byte-array	No	The key whose value is replaced.
value	byte-array	No	The new value to be associated with the specified key.
expiryPolicy	byte-array	Yes	Expiry policy for the entry. Byte-array which is serialized from an object implementing javax.cache.expiry.ExpiryPolicy interface.
completionId	int32	No	User generated id which shall be received as a field of the cache event upon completion of the request in the cluster.

Response Message

Type Id : 0x0069

The old value previously assigned to the given key.

Name	Type	Nullable
response	byte-array	Yes

Cache.GetConfig

This message is idempotent.

Request Message

Type Id : 0x150d
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the cache with prefix.
simpleName	string	No	Name of the cache without prefix.

Response Message

Type Id : 0x0069

The cache configuration. Byte-array which is serialized from an object implementing javax.cache.configuration.Configuration interface.

Name	Type	Nullable
response	byte-array	Yes

Cache.Get

Retrieves the mapped value of the given key using a custom javax.cache.expiry.ExpiryPolicy. If no mapping exists null is returned. If the cache is configured for read-through operation mode, the underlying configured javax.cache.integration.CacheLoader might be called to retrieve the value of the key from any kind of external resource. This message is idempotent.

Request Message

Type Id : 0x150e
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the cache.
key	byte-array	No	The key whose mapped value is to be returned.
expiryPolicy	byte-array	Yes	Expiry policy for the entry. Byte-array which is serialized from an object implementing javax.cache.expiry.ExpiryPolicy interface.

Response Message

Type Id : 0x0069

The value assigned to the given key, or null if not assigned.

Name	Type	Nullable
response	byte-array	Yes

Cache.Iterate

The ordering of iteration over entries is undefined. During iteration, any entries that are a). read will have their appropriate CacheEntryReadListeners notified and b). removed will have their appropriate CacheEntryRemoveListeners notified. java.util.Iterator#next() may return null if the entry is no longer present, has expired or has been evicted.

Request Message

Type Id : 0x150f
Partition Id : the value passed in partitionId field

Name	Type	Nullable	Description
name	string	No	Name of the cache.
partitionId	int32	No	The partition id which owns this cache store.
tableIndex	int32	No	The slot number (or index) to start the iterator
batch	int32	No	The number of items to be batched

Response Message

Type Id : 0x0074

last index processed and list of data

Name	Type	Nullable
tableIndex	int32	No
keys	array of byte-array	No

Cache.ListenerRegistration

Request Message

Type Id : 0x1510
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the cache.
listenerConfig	byte-array	No	The listener configuration. Byte-array which is serialized from an object implementing javax.cache.configuration.CacheEntryListenerConfiguration
shouldRegister	boolean	No	true if the listener is being registered, false if the listener is being unregistered.
address	Address	No	The address of the member server for which the listener is being registered for.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Cache.LoadAll

Request Message

Type Id : 0x1511
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the cache.
keys	array of byte-array	No	the keys to load
replaceExistingValues	boolean	No	when true existing values in the Cache will be replaced by those loaded from a CacheLoader

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Cache.ManagementConfig

This message is idempotent.

Request Message

Type Id : 0x1512
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the cache.
isStat	boolean	No	true if enabling statistics, false if enabling management.

enabled	boolean	No	true if enabled, false to disable.
address	Address	No	the address of the host to enable.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Cache.PutIfAbsent

Associates the specified key with the given value if and only if there is not yet a mapping defined for the specified key. If the cache is configured for write-through operation mode, the underlying configured javax.cache.integration.CacheWriter might be called to store the value of the key to any kind of external resource.

Request Message

Type Id : 0x1513
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the cache.
key	byte-array	No	The key that is associated with the specified value.
value	byte-array	No	The value that has the specified key associated with it.
expiryPolicy	byte-array	Yes	The custom expiry policy for this operation. A null value is equivalent to put(Object, Object).
completionId	int32	No	User generated id which shall be received as a field of the cache event upon completion of the request in the cluster.

Response Message

Type Id : 0x0065

true if a value was set, false otherwise.

Name	Type	Nullable
response	boolean	No

Cache.Put

Request Message

Type Id : 0x1514
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the cache.
key	byte-array	No	The key that has the specified value associated with it.
value	byte-array	No	The value to be associated with the key.
expiryPolicy	byte-array	Yes	Expiry policy for the entry. Byte-array which is serialized from an object implementing javax.cache.expiry.ExpiryPolicy interface.

get	boolean	No	boolean flag indicating if the previous value should be retrieved.
completionId	int32	No	User generated id which shall be received as a field of the cache event upon completion of the request in the cluster.

Response Message

Type Id : 0x0069

The value previously assigned to the given key, or null if not assigned.

Name	Type	Nullable
response	byte-array	Yes

Cache.RemoveEntryListener

This message is idempotent.

Request Message

Type Id : 0x1515
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the cache.
registrationId	string	No	The id assigned during the registration for the listener which shall be removed.

Response Message

Type Id : 0x0065

true if the listener is de-registered, false otherwise

Name	Type	Nullable
response	boolean	No

Cache.RemoveInvalidationListener

This message is idempotent.

Request Message

Type Id : 0x1516
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the cache.
registrationId	string	No	The id assigned during the registration for the listener which shall be removed.

Response Message

Type Id : 0x0065

true if the listener is de-registered, false otherwise

Name	Type	Nullable
response	boolean	No

Cache.Remove

Atomically removes the mapping for a key only if currently mapped to the given value.

Request Message

Type Id : 0x1517
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the cache.
key	byte-array	No	key whose mapping is to be removed from the cache
currentValue	byte-array	Yes	value expected to be associated with the specified key.
completionId	int32	No	User generated id which shall be received as a field of the cache event upon completion of the request in the cluster.

Response Message

Type Id : 0x0065

returns false if there was no matching key

Name	Type	Nullable
response	boolean	No

Cache.Replace

Atomically replaces the currently assigned value for the given key with the specified newValue if and only if the currently assigned value equals the value of oldValue using a custom javax.cache.expiry.ExpiryPolicy
If the cache is configured for write-through operation mode, the underlying configured javax.cache.integration.CacheWriter might be called to store the value of the key to any kind of external resource.

Request Message

Type Id : 0x1518
Partition Id : Murmur hash of key % partition count

Name	Type	Nullable	Description
name	string	No	Name of the cache.
key	byte-array	No	The key whose value is replaced.
oldValue	byte-array	Yes	Old value to match if exists before removing. Null means "don't try to remove"
newValue	byte-array	No	The new value to be associated with the specified key.
expiryPolicy	byte-array	Yes	Expiry policy for the entry. Byte-array which is serialized from an object implementing javax.cache.expiry.ExpiryPolicy interface.
completionId	int32	No	User generated id which shall be received as a field of the cache event upon completion of the request in the cluster.

Response Message

Type Id : 0x0069

The replaced value.

Name	Type	Nullable
response	byte-array	Yes

Cache.Size

Total entry count This message is idempotent.

Request Message

Type Id : 0x1519
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the cache.

Response Message

Type Id : 0x0066

total entry count

Name	Type	Nullable
response	int32	No

Cache.AddPartitionLostListener

Adds a CachePartitionLostListener. The addPartitionLostListener returns a registration ID. This ID is needed to remove the CachePartitionLostListener using the #removePartitionLostListener(String) method. There is no check for duplicate registrations, so if you register the listener twice, it will get events twice.Listeners registered from HazelcastClient may miss some of the cache partition lost events due to design limitations.

Request Message

Type Id : 0x151a
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the cache
localOnly	boolean	No	if true only node that has the partition sends the request, if false sends all partition lost events.

Response Message

Type Id : 0x0068

returns the registration id for the CachePartitionLostListener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00d6

Name	Type	Nullable	Description
partitionId	int32	No	
uuid	string	No	

Cache.RemovePartitionLostListener

Removes the specified cache partition lost listener. Returns silently if there is no such listener added before This message is idempotent.

Request Message

Type Id : 0x151b
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	Name of the Cache
registrationId	string	No	ID of registered listener.

Response Message

Type Id : 0x0065

true if registration is removed, false otherwise.

Name	Type	Nullable
response	boolean	No

Cache.PutAll

Request Message

Type Id : 0x151c
Partition Id : -1

Name	Type	Nullable	Description
name	string	No	name of the cache
entries	array of key-value byte array pair	No	entries to be put as batch
expiryPolicy	byte-array	Yes	expiry policy for the entry. Byte-array which is serialized from an object implementing {
completionId	int32	No	user generated id which shall be received as a field of the cache event upon completion of the request in the cluster.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

XATransactional

XATransactional.ClearRemote

Request Message

Type Id : 0x1601
Partition Id : Murmur hash of xid % partition count

Name	Type	Nullable	Description
xid	Transaction Id	No	Java XA transaction id as defined in interface javax.transaction.xa.Xid.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

XATransactional.CollectTransactions

Request Message

Type Id : 0x1602
Partition Id : -1

Header only request message, no message body exist.

Response Message

Type Id : 0x006a

Array of Xids.

Name	Type	Nullable
response	array of byte-array	No

XATransactional.Finalize

Request Message

Type Id : 0x1603
Partition Id : Murmur hash of xid % partition count

Name	Type	Nullable	Description

xid	Transaction Id	No	Java XA transaction id as defined in interface javax.transaction.xa.Xid.
isCommit	boolean	No	If true, the transaction is committed else transaction is rolled back.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

XATransactional.Commit

Request Message

Type Id : 0x1604
Partition Id : -1

Name	Type	Nullable	Description
transactionId	string	No	The internal Hazelcast transaction id.
onePhase	boolean	No	If true, the prepare is also done.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

XATransactional.Create

Request Message

Type Id : 0x1605
Partition Id : -1

Name	Type	Nullable	Description
xid	Transaction Id	No	Java XA transaction id as defined in interface javax.transaction.xa.Xid.
timeout	int64	No	The timeout in seconds for XA operations such as prepare, commit, rollback.

Response Message

Type Id : 0x0068

The transaction unique identifier.

Name	Type	Nullable
response	string	No

XATransactional.Prepare

Request Message

Type Id : 0x1606
Partition Id : -1

Name	Type	Nullable	Description
transactionId	string	No	The id of the transaction to prepare.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

XATransactional.Rollback

Request Message

Type Id : 0x1607
Partition Id : -1

Name	Type	Nullable	Description
transactionId	string	No	The id of the transaction to rollback.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Transactional

Transactional.Commit

Request Message

Type Id : 0x1701
Partition Id : -1

Name	Type	Nullable	Description
transactionId	string	No	The internal Hazelcast transaction id.
threadId	int64	No	The thread id for the transaction.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

Transactional.Create

Request Message

Type Id : 0x1702
Partition Id : -1

Name	Type	Nullable	Description
timeout	int64	No	The maximum allowed duration for the transaction operations.
durability	int32	No	The durability of the transaction
transactionType	int32	No	Identifies the type of the transaction. Possible values are: 1 (Two phase): The two phase commit is more than the classic two phase commit (if you want a regular two phase commit, use local). Before it commits, it copies the commit-log to other members, so in case of member failure, another member can complete the commit. 2 (Local): Unlike the name suggests, local is a two phase commit. So first all cohorts are asked to prepare if everyone agrees then all cohorts are asked to commit. The problem happens when during the commit phase one or more members crash, that the system could be left in an inconsistent state.
threadId	int64	No	The thread id for the transaction.

Response Message

Type Id : 0x0068

The transaction id for the created transaction.

Name	Type	Nullable
response	string	No

Transactional.Rollback

Request Message

Type Id : 0x1703
Partition Id : -1

Name	Type	Nullable	Description
transactionId	string	No	The internal Hazelcast transaction id.
threadId	int64	No	The thread id for the transaction.

Response Message

Type Id : 0x0064

Header only response message, no message body exist.

EnterpriseMap

EnterpriseMap.PublisherCreateWithValue

This message is idempotent.

Request Message

Type Id : 0x1801
Partition Id : -1

Name	Type	Nullable	Description
mapName	string	No	Name of the map.
cacheName	string	No	Name of the cache for query cache.
predicate	byte-array	No	The predicate to filter events which will be applied to the QueryCache.
batchSize	int32	No	The size of batch. After reaching this minimum size, node immediately sends buffered events to QueryCache.
bufferSize	int32	No	Maximum number of events which can be stored in a buffer of partition.
delaySeconds	int64	No	The minimum number of delay seconds which an event waits in the buffer of node.
populate	boolean	No	Flag to enable/disable initial population of the QueryCache.
coalesce	boolean	No	Flag to enable/disable coalescing. If true, then only the last updated value for a key is placed in the batch, otherwise all changed values are included in the update.

Response Message

Type Id : 0x0075

Array of key-value pairs.

Name	Type	Nullable
response	array of key-value byte array pair	No

EnterpriseMap.PublisherCreate

This message is idempotent.

Request Message

Type Id : 0x1802
Partition Id : -1

Name	Type	Nullable	Description
mapName	string	No	Name of the map.
cacheName	string	No	Name of query cache.
predicate	byte-array	No	The predicate to filter events which will be applied to the QueryCache.
batchSize	int32	No	The size of batch. After reaching this minimum size, node immediately sends buffered events to QueryCache.
bufferSize	int32	No	Maximum number of events which can be stored in a buffer of partition.
delaySeconds	int64	No	The minimum number of delay seconds which an event waits in the buffer of node.
populate	boolean	No	Flag to enable/disable initial population of the QueryCache.
coalesce	boolean	No	Flag to enable/disable coalescing. If true, then only the last updated value for a key is placed in the batch, otherwise all changed values are included in the update.

Response Message

Type Id : 0x006a

Array of keys.

Name	Type	Nullable
response	array of byte-array	No

EnterpriseMap.MadePublishable

This message is idempotent.

Request Message

Type Id : 0x1803
Partition Id : -1

Name	Type	Nullable	Description
mapName	string	No	Name of the map.
cacheName	string	No	Name of query cache.

Response Message

Type Id : 0x0065

True if successfully set as publishable, false otherwise.

Name	Type	Nullable
response	boolean	No

EnterpriseMap.AddListener

Request Message

Type Id : 0x1804
Partition Id : -1

Name	Type	Nullable	Description
listenerName	string	No	Name of the MapListener which will be used to listen this QueryCache
localOnly	boolean	No	if true fires events that originated from this node only, otherwise fires all events

Response Message

Type Id : 0x0068

Registration id for the listener.

Name	Type	Nullable
response	string	No

Event Message

Type Id : 0x00d4

Name	Type	Nullable	Description
data	Query Cache Event Data	No	Query cache map event data.

Event Message

Type Id : 0x00d5

Name	Type	Nullable	Description
events	array of Query Cache Event Data	No	Array of query cache events
source	string	No	Source of the event.
partitionId	int32	No	The partition id for the query cache.

EnterpriseMap.SetReadCursor

This method can be used to recover from a possible event loss situation.
This method tries to make consistent the data in this `QueryCache` with the data in the underlying `IMap` by replaying the events after last consistently received ones. As a result of this replaying logic, same event may appear more than once to the `QueryCache` listeners.
This method returns `false` if the event is not in the buffer of event publisher side. That means recovery is not possible.

Request Message

Type Id : 0x1805
Partition Id : Murmur hash of associated key % partition count

Name	Type	Nullable	Description
mapName	string	No	Name of the map.
cacheName	string	No	Name of query cache.
sequence	int64	No	The cursor position of the accumulator to be set.

Response Message

Type Id : 0x0065

True if the cursor position could be set, false otherwise.

Name	Type	Nullable
response	boolean	No

EnterpriseMap.DestroyCache

Request Message

Type Id : 0x1806
Partition Id : -1

Name	Type	Nullable	Description
mapName	string	No	Name of the map.
cacheName	string	No	Name of query cache.

Response Message

Type Id : 0x0065

True if all cache is destroyed, false otherwise.

Name	Type	Nullable
response	boolean	No

Note: All operation defined for the Map Object can also be executed against the EnterpriseMap Object.

Ringbuffer

Ringbuffer.Size

Returns number of items in the ringbuffer. If no ttl is set, the size will always be equal to capacity after the head completed the first looparound the ring. This is because no items are getting retired.

Request Message

Type Id : 0x1901
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Ringbuffer

Response Message

Type Id : 0x0067

the size

Name	Type	Nullable
response	int64	No

Ringbuffer.TailSequence

Returns the sequence of the tail. The tail is the side of the ringbuffer where the items are added to.
The initial value of the tail is -1.

Request Message

Type Id : 0x1902
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Ringbuffer

Response Message

Type Id : 0x0067

the sequence of the tail

Name	Type	Nullable
response	int64	No

Ringbuffer.HeadSequence

Returns the sequence of the head. The head is the side of the ringbuffer where the oldest items in the ringbuffer are found. If the RingBuffer is empty, the head will be one more than the tail.
The initial value of the head is 0 (1 more than tail).

Request Message

Type Id : 0x1903
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Ringbuffer

Response Message

Type Id : 0x0067

the sequence of the head

Name	Type	Nullable
response	int64	No

Ringbuffer.Capacity

Returns the capacity of this Ringbuffer.

Request Message

Type Id : 0x1904
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Ringbuffer

Response Message

Type Id : 0x0067

the capacity

Name	Type	Nullable
response	int64	No

Ringbuffer.RemainingCapacity

Returns the remaining capacity of the ringbuffer. The returned value could be stale as soon as it is returned.
If ttl is not set, the remaining capacity will always be the capacity.

Request Message

Type Id : 0x1905
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Ringbuffer

Response Message

Type Id : 0x0067

the remaining capacity

Name	Type	Nullable
response	int64	No

Ringbuffer.Add

Adds an item to the tail of the Ringbuffer. If there is space in the ringbuffer, the call
will return the sequence of the written item. If there is no space, it depends on the overflow policy what happens:
OverflowPolicy OVERWRITE we just overwrite the oldest item in the ringbuffer and we violate the ttl
OverflowPolicy FAIL we return -1. The reason that FAIL exist is to give the opportunity to obey the ttl.

This sequence will always be unique for this Ringbuffer instance so it can be used as a unique id generator if you are publishing items on this Ringbuffer. However you need to take care of correctly determining an initial id when any node uses the ringbuffer for the first time. The most reliable way to do that is to write a dummy item into the ringbuffer and use the returned sequence as initial id. On the reading side, this dummy item should be discard. Please keep in mind that this id is not the sequence of the item you are about to publish but from a previously published item. So it can't be used to find that item.

Request Message

Type Id : 0x1906
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Ringbuffer
overflowPolicy	int32	No	the OverflowPolicy to use.
value	byte-array	No	to item to add

Response Message

Type Id : 0x0067

the sequence of the added item, or -1 if the add failed.

Name	Type	Nullable

response	int64	No

Ringbuffer.ReadOne

Reads one item from the Ringbuffer. If the sequence is one beyond the current tail, this call blocks until an item is added. This method is not destructive unlike e.g. a queue.take. So the same item can be read by multiple readers or it can be read multiple times by the same reader. Currently it isn't possible to control how long this call is going to block. In the future we could add e.g. tryReadOne(long sequence, long timeout, TimeUnit unit).

Request Message

Type Id : 0x1908
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Ringbuffer
sequence	int64	No	the sequence of the item to read.

Response Message

Type Id : 0x0069

the read item

Name	Type	Nullable
response	byte-array	Yes

Ringbuffer.AddAll

Adds all the items of a collection to the tail of the Ringbuffer. A addAll is likely to outperform multiple calls to add(Object) due to better io utilization and a reduced number of executed operations. If the batch is empty, the call is ignored. When the collection is not empty, the content is copied into a different data-structure. This means that: after this call completes, the collection can be re-used. the collection doesn't need to be serializable. If the collection is larger than the capacity of the ringbuffer, then the items that were written first will be overwritten. Therefor this call will not block. The items are inserted in the order of the Iterator of the collection. If an addAll is executed concurrently with an add or addAll, no guarantee is given that items are contiguous. The result of the future contains the sequenceId of the last written item

Request Message

Type Id : 0x1909
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Ringbuffer
valueList	array of byte-array	No	the batch of items to add
overflowPolicy	int32	No	the overflowPolicy to use

Response Message

Type Id : 0x0067

the ICompletableFuture to synchronize on completion.

Name	Type	Nullable
response	int64	No

Ringbuffer.ReadMany

Reads a batch of items from the Ringbuffer. If the number of available items after the first read item is smaller
than the maxCount, these items are returned. So it could be the number of items read is smaller than the maxCount.
If there are less items available than minCount, then this call blacks. Reading a batch of items is likely to
perform better because less overhead is involved. A filter can be provided to only select items that need to be read.
If the filter is null, all items are read. If the filter is not null, only items where the filter function returns
true are returned. Using filters is a good way to prevent getting items that are of no value to the receiver.
This reduces the amount of IO and the number of operations being executed, and can result in a significant performance improvement.

Request Message

Type Id : 0x190a
Partition Id : Murmur hash of name % partition count

Name	Type	Nullable	Description
name	string	No	Name of the Ringbuffer
startSequence	int64	No	the startSequence of the first item to read
minCount	int32	No	the minimum number of items to read.
maxCount	int32	No	the maximum number of items to read.
filter	byte-array	Yes	Filter is allowed to be null, indicating there is no filter.

Response Message

Type Id : 0x0073

a future containing the items read.

Name	Type	Nullable
readCount	int32	No
items	array of byte-array	No

Glossary

Terminology	Definition
client	Any Hazelcast native client implementation
server/member	A Hazelcast cluster member
protocol	Hazelcast client-server communication protocol
serialization	Hazelcast internal implementation of serialization used to encode an object into a byte array and decode a byte array into an object
protocol-version	The version of the protocol starting at 1
fragmentation	Splitting a large message into pieces for transmission
Reassembly	Combining the message parts (fragments) to form the actual large message on reception
Cluster	A virtual environment formed by Hazelcast members communicating with each other