# QMF Map Message Protocol

# QMFv2 Map Message Protocol

# Introduction

This document describes the design of a proposed protocol for QMF based on map-messages (offered by the new C++ and Python APIs as well as the existing JMS API).

If adopted, this new protocol will change the formats of the messages used by QMF components to communicate. It will also change some of the message exchange patterns. It will not significantly impact the console and agent APIs and is intended to operate with applications that use the current QMF APIs.

Some highlights of the new design:

- · Current QMF message bodies are in packed binary formats. While quite efficient, this style of formatting makes it difficult to make changes to the format and content for new features. The proposed format is based on encoded maps (a.k.a. dictionaries, field-tables) which are very easily extended and require less context to be useful.
- QMF currently requires the message broker to participate in the QMF protocol. The proposed protocol removes this requirement and will run properly on any AMQP message broker.
- QMF Agents currently publish periodic updates of their managed content to a globally accessible topic. This has security implications with regard to access to data. This is also inflexible in that updates to all data are sent at the same intervals. The proposed protocol removes the global publishing of data and introduces a subscription-query whereby a console may request that an agent publish certain data at a certain interval to an indicated target. Such requests can be subject to access control and may be focused on only the data that is needed for a particular application. •
  - The proposed protocol allows for more general use of data. For example:
    - Free-form data, that has no object-identifier nor schema, can be transferred. This is useful for complex queries (joins, reports, etc.). <sup>o</sup> Methods can be invoked against an agent in the absence of a managed object.

# **QMF** Protocol

## Use of Message Headers

#### **Standard Message Properties**

| Message<br>Property | Use   |
|---------------------|---|
| correlation-id      | Used in request/response/indication sets to correlate responses and indications to their request. |
| reply-to            | Used in requests to indicate the address for the response.  |
| content-type        | 'amqp/map' or 'amqp/list'   |
| user-id             | Supplied in a request if authentication/authorization at the agent is appropriate.                |
| app-id              | 'qmf2'  |

#### **Custom Application Headers**

| Application<br>Header Key | Use   |
|---------------------------|---|
| method                    | 'request', 'response', or 'indication'. This field describes the message's role in a particular message-exchange pattern.   |
| partial                   | Void. If this field is present, it indicates that the message does not contain the complete request or response (i.e. another message will follow using the same opcode and correlation-id) |
| qmf.opcode                | QMF-specific operation code (see list below). The opcode defines what content, if any, is to be found in the message body.  |
| qmf.content               | If the opcode is a data indication, this field indicates what kind of data will be found in the message body.   |
| qmf.agent                 | If this message is a data indication sent by an agent, this field contains the agent's name.  |

#### **QMF OpCodes**

| qmf.<br>opcode<br>field | Message<br>Body<br>Data<br>Type | Sent<br>By | Sent<br>To | Description  |
|-------------------------|---------------------------------|------------|------------|--|
| _exception              | QMF_DATA                        | Agent      | reply-to   | This general-purpose message can be sent by an agent in response to any request (query, subscription, method) if the request could not be completed for any reason. The QMF_DATA in the message body contains details of why the failure occurred. |

| _agent_loca<br>te_request             | QMF_QUER<br>Y_PREDICA<br>TE            | Console | QMF<br>Topic         | A console may send an agent-locate-request in order to reach all available agents. The predicate may be used to limit the set of agents that will respond to the request.  |  |
|---------------------------------------|--|---------|----------------------|--|--|
| _agent_loca<br>te_response            | QMF_DATA                               | Agent   | reply-to             | This is a response to an agent-locate-request. An agent will send an agent-locate-response if it received an agent-locate-request with a predicate that matches its characteristics.   |  |
| _agent_hea<br>rtbeat_indic<br>ation   | QMF_DATA                               | Agent   | Торіс                | Each agent periodically sends a heartbeat message to a topic to indicate that it is alive and connected. The content of the heartbeat message is the list of the agent's characteristics.  |  |
| _query_req<br>uest                    | QMF_QUER<br>Y                          | Console | Agent                | A console sends a query to an agent to request that the agent send data to the requester.  |  |
| _query_res<br>ponse                   | List of <qmf.<br>content&gt;</qmf.<br> | Agent   | reply-to             | The response to a query sent by a console.   |  |
| _subscribe_<br>request                | QMF_SUBS<br>CRIBE                      | Console | Agent                | A console sends a subscribe-request to an agent to receive data matched by a query. A subscription differs from a query request in continues to send updated information to the console when the data changes.   |  |
| _subscribe_<br>response               | QMF_SUBS<br>CRIPTION                   | Agent   | reply-to             | When an agent receives a subscribe-request, it sends a subscribe-response granting (or refusing) the subscription. Should the<br>subscription succeed, the response will contain an identifier for the subscription assigned by the Agent. Thereafter, it will send data-<br>indication messages on the same correlation-id with updates when they happen or periodically. The first data-indication message sent by<br>the agent will contain all matching data, subsequent data-indications will contain only those matching data that has changed since the last<br>update. |  |
| _subscribe_<br>cancel_indic<br>ation  | QMF_SUBS<br>CRIPTION_I<br>D            | Console | Agent                | A console can request that a subscription it created be immediately cancelled. This message must have the same correlation-id as the original request, and contain the subscription identifier as assigned by the Agent.   |  |
| _subscribe_<br>refresh_indi<br>cation | QMF_SUBS<br>CRIPTION_I<br>D            | Console | Agent                | A console can keep a subscription alive by periodically refreshing it by sending a subscribe-refresh-indication. This message must have the same correlation-id as the original request, and contain the subscription identifier as assigned by the Agent.   |  |
| _data_indic ation                     | List of <qmf.<br>content&gt;</qmf.<br> | Agent   | reply-to<br>or topic | A data indication is sent by an Agent when 1) subscription data has changed and needs to be published, 2) an event has occurred and event data is being published, and 3) any other time an agent wants to send unsolicited data.  |  |
| _method_re<br>quest                   | QMF_METH<br>OD_CALL                    | Console | Agent                | A console may invoke a method on an object managed by an agent. It may also invoke a method directly on the agent if appropriate. This message contains the method call including the input arguments.   |  |
| _method_re<br>sponse                  | QMF_METH<br>OD RESULT                  | Agent   | reply-to             | A method call always results in a single method result. This message carries either the output arguments from a successful method call or it holds an exception to describe a failure.   |  |

## **QMF Content Types**

| qmf.content<br>field | Data Type        | Description                                     |
|----------------------|------------------|---|
| _schema_id           | SCHEMA_ID        | Schema class identifier                         |
| _schema_class        | SCHEMA_CLAS<br>S | Schema class definition                         |
| _object_id           | OBJECT_ID        | Managed object identifier                       |
| _data                | QMF_DATA         | Data, managed and/or described or free-<br>form |
| _event               | QMF_EVENT        | Event   |
| _query               | QMF_QUERY        | Query   |

# Message Body Map Formats

## SCHEMA\_ID

```
SCHEMA_ID := { _package_name: STRING,
    _class_name: STRING,
    _type: '_data' | '_event',
    _hash: UUID
 }
```

| Field             | Optional | Description  |
|-------------------|----------|--|
| _package_nam<br>e | no       | Package name (namespace) for the described class         |
| _class_name       | no       | Name of the described class                              |
| _type             | no       | Class type: data or event                                |
| _hash             | yes      | Hash (uuid) to distinguish different versions of a class |

#### SCHEMA\_CLASS

```
SCHEMA_CLASS := { _schema_id: SCHEMA_ID,
   _desc: STRING,
   _default_severity: NUMBER,
   _properties: [SCHEMA_PROPERTY, ...],
   _methods: [SCHEMA_METHOD, ...]
}
```

| Field                 | Optional | Description   |
|-----------------------|----------|---|
| _schema_id            | no       | Identifier for this schema class  |
| _desc                 | yes      | Description of the schema class   |
| _default_severit<br>y | yes      | Default severity for an event class                                     |
| _properties           | no       | List of SCHEMA_PROPERTY elements describing the properties of the class |
| _methods              | no       | List of SCHEMA_METHOD elements describing the methods of the class      |

#### SCHEMA\_PROPERTY

```
SCHEMA_PROPERTY := { _name: STRING,
   _type: QMF_TYPE,
   _index: BOOL,
   _access: 'RO' | 'RC' | 'RW',
   _unit: STRING,
   _min: NUMBER,
   _max: NUMBER,
   _dir: 'I' | 'O' | 'IO',
   _desc: STRING,
   _references: SCHEMA_ID,
   _subtype: QMF_SUBTYPE
 }
```

| Field           | Optional | Description  |
|-----------------|----------|--|
| _name           | no       | The name of the property   |
| _type           | no       | The QMF data type of this property   |
| _index          | yes      | True iff this property is an index of an object. Default is False.   |
| _access         | yes      | The remote access rules for this property:<br>RO => Read Only (default if not specified)<br>RC => Read Create<br>RW => Read Write  |
| _unit           | yes      | Annotation. Units of measure for numeric values  |
| _min            | yes      | Minimum numeric value  |
| _max            | yes      | Maximum numeric value  |
| _maxlen         | yes      | Maximum length of a variable length value (in octets)  |
| _dir            | yes      | Used only for method arguments. Direction of transfer:<br>I => Input (caller to callee)<br>O => Output (callee to caller)<br>IO => Both  |
| _desc           | yes      | Annotation. Description of the property  |
| _referen<br>ces | yes      | If the type is a reference to another managed object, this field may be used to specify the required class for that object   |
| _subtyp<br>e    | yes      | May be used to further specify the meaning of the value of this field. For example, a number may actually be a timestamp or a duration. A string may be a reference to another object, or a URL. |

```
QMF_TYPE := 'TYPE_VOID'
'TYPE_BOOL'
'TYPE_INT'
'TYPE_FLOAT'
'TYPE_STRING'
'TYPE_MAP'
'TYPE_LIST'
'TYPE_UUID'
```

## QMF\_SUBTYPE

```
QMF_SUBTYPE := 'reference' |
    'url' |
    'timestamp' |
    'duration'
```

#### SCHEMA\_METHOD

| SCHEMA_METHOD | := { | _name:      | STRING,           |         |
|---------------|------|-------------|-------------------|---------|
|               |      | _desc:      | STRING,           |         |
|               |      | _arguments: | [SCHEMA_PROPERTY, | • • • • |
|               | }    |             |                   |         |

| Field          | Optional | Description   |
|----------------|----------|---|
| _name          | no       | The name of the method  |
| _desc          | yes      | Annotation. Description of this method                                |
| _argument<br>s | no       | List of SCHEMA_PROPERTY elements that describe the method's arguments |

#### QMF\_METHOD\_CALL

| Field            | Optional | Description  |
|------------------|----------|--|
| _object_id       | yes      | The identity of the managed object receiving the method call. If not supplied, this method applies generally to the agent. |
| _method_nam<br>e | no       | The name of the method   |
| _arguments       | yes      | The input arguments, if any  |
| _subtypes        | yes      | Subtype information for the input arguments, if any  |

## QMF\_METHOD\_RESULT

QMF\_METHOD\_RESULT := { \_arguments: { EACH\_KEY: VALUE }, \_subtypes: { EACH\_KEY: STRING }
}

| Field          | Optional | Description  |
|----------------|----------|--|
| _argument<br>s | yes      | Output arguments from a successful method call, if any |
| _subtypes      | yes      | Subtype information for the output arguments, if any   |

#### QMF\_DATA

```
QMF_DATA := { _schema_id: SCHEMA_ID,
    _object_id: OBJECT_ID,
    _values: { EACH_KEY: VALUE },
    _subtypes: { EACH_KEY: STRING }
  }
```

| Field          | Optional | Description   |
|----------------|----------|---|
| _schema_i<br>d | yes      | If this data is "described", this field references the schema class that describes the data.                      |
| _object_id     | yes      | If this data is "managed", this field provides the identifier that can be used to address this managed object.    |
| _values        | no       | The map of values keyed by their property names   |
| _subtypes      | yes      | Per-property subtypes that may be used to provide more information about the meaning of a value than its QMF_TYPE |

# OBJECT\_ID

```
OBJECT_ID := { _agent_name: STRING,
      _agent_epoch: NUMBER,
      _object_name: STRING
  }
```

| Field            | Optional | Description   |  |  |
|------------------|----------|---|--|--|
| _agent_<br>name  | yes      | Name of the agent that is managing the referenced data  |  |  |
| _agent_<br>epoch | yes      | Numeric epoch of the agent process. This number is managed by the agent and is incremented each time the agent process starts. This field is only present for <i>transient</i> object IDs that must not be the same for a given object across an agent restart. <i>Persistent</i> object IDs must not include this field. |  |  |
| _object_<br>name | no       | Name of the data that uniquely identifies the data within the context of the agent.   |  |  |

#### QMF\_QUERY

```
QMF_QUERY := { _what: QMF_QUERY_TARGET,
    _where: QMF_QUERY_PREDICATE,
    _object_id: OBJECT_ID,
    _schema_id: SCHEMA_ID
 }
```

| Field          | Optional | Description   |
|----------------|----------|---|
| _what          | no       | Identifies the kind of data being queried                   |
| _where         | yes      | Query predicate to limit the number of results of the query |
| _object_id     | yes      | Identifier of a single object being queried                 |
| _schema_i<br>d | yes      | Identifier of a single schema being queried                 |

## QMF\_QUERY\_TARGET

```
QMF_QUERY_TARGET := 'SCHEMA_ID' |
'SCHEMA' |
'OBJECT_ID' |
'OBJECT'
```

## QMF\_QUERY\_PREDICATE

# QMF\_SUBSCRIBE

| QMF_SUBSCRIBE := { _query: | QMF_QUERY, |
|----------------------------|------------|
| _duration:                 | NUMBER,    |
| _interval:                 | NUMBER     |
| }                          |            |

| Field         | Optional | Description   |
|---------------|----------|---|
| _query        | no       | The query that defines the set of data being subscribed to  |
| _duratio<br>n | yes      | The requested time (in seconds) after which this subscription will be automatically canceled. If a <b>subscribe_refresh_indication</b> is received by the agent running this query, this time interval will start over. |
| _interval     | yes      | The request time (in milliseconds) between periodic updates of data in this subscription. The agent may place a minimum on this interval.   |

## QMF\_SUBSCRIPTION

| <pre>QMF_SUBSCRIPTION := { _subscription_id:<br/>_duration: NUMBER,<br/>_interval: NUMBER,<br/>}</pre> | STRING, |
|--|---------|
|  |         |

| Field                | Optional | Description   |  |
|----------------------|----------|---|--|
| _subscripti<br>on_id | yes      | Assigned by the Agent when replying to a successful subscription request. Must be supplied by the Console when sending a subscription refresh or cancel to the Agent for this subscription. |  |
| _duration            | no       | The time (in seconds) after which this subscription will be automatically canceled.   |  |
| _interval            | no       | The time (in milliseconds) between periodic updates of data in this subscription.   |  |

# QMF\_SUBSCRIPTION\_ID

|  | QMF_SUBSCRIPTION_I | := | { | _subscription_id: | STRING } |
|--|--------------------|----|---|-------------------|----------|
|--|--------------------|----|---|-------------------|----------|

| Field                | Optional | Description   |
|----------------------|----------|---|
| _subscription_i<br>d | no       | Supplied by the Console when sending a subscription refresh or cancel to the Agent for this subscription. |