

AMQ RFC011

AMQP Reply Codes

version 0.1a

iMatix Corporation <amq@imatix.com>

Copyright (c) 2004 JPMorgan

Revised: 2005/02/10

Contents

1	Cover	1
1.1	State of this Document	1
1.2	Copyright Notice	1
1.3	Authors	1
1.4	Abstract	1
2	Introduction	2
2.1	Problem Statement	2
2.2	Argumentation	2
2.3	Basic Proposal	2
3	Design Proposal	3
3.1	Definitions and References	3
3.2	Objectives	3
3.3	Detailed Proposal	3
3.4	Alternatives	4
3.5	Security Considerations	4
4	Comments on this Document	5
4.1	Date, name	5

1 Cover

1.1 State of this Document

This document is a request for comments. Distribution of this document is currently limited to iMatix and JPMorgan internal use.

This document describes a work in progress. This document is a formal standard. This document is ready for review.

1.2 Copyright Notice

This document is copyright (c) 2004 JPMorgan Inc.

1.3 Authors

This document was written by Pieter Hintjens <ph@imatix.com>.

1.4 Abstract

This document defines the standard set of 3-digit reply codes used by AMQ implementations including OpenAMQ. The reply codes are based on the IETF standard model as described in IETF RFC2821 and RFC2822.

2 Introduction

2.1 Problem Statement

Reply codes allow a server to tell a client whether an operation succeeded and if not, what the status of the operation is. A well-known example is the HTTP reply code 404 meaning "resource not found".

2.2 Argumentation

Reply codes must be clearly defined, and processable by software. The IETF standard has evolved over some time and is today a recognised solution to the problem of error reporting. The 3-digit reply code distinguishes cleanly between a well-defined set of outcomes that cover the needs of modern protocol implementations. We do not need to reinvent a solution to this problem, only define our implementation of it.

2.3 Basic Proposal

We use the IETF standard format for reply codes as described in IETF RFC821. A reply code uses three digits, and the first digit provides the main feedback as to whether and how an operation completed. The second and third digits provide additional information. The reply codes can be processed by client applications without full knowledge of their meaning.

3 Design Proposal

3.1 Definitions and References

The IETF RFC2821 and RFC2822 (SMTP) provides the theory and definition of reply codes.

3.2 Objectives

We define a set of reply codes that are consistent and compatible with the model defined in RFC821.

3.3 Detailed Proposal

We use a standard 3-digit reply code. The reply code is constructed as follows:

First digit (completion) reports whether the request succeeded or not:

- 1** Ready to be performed, pending some confirmation.
- 2** Successful.
- 3** Ready to be performed, pending more information.
- 4** Failed, but may succeed later.
- 5** Failed, requires intervention.
- 6-9** Reserved for future use.

Second digit (category) provides more information on failures:

- 0** Error in syntax.
- 1** The reply provides general information.
- 2** Problem with session or connection.
- 3** Problem with security.
- 4** Problem with implementation.
- 5-9** Reserved for future use.

Third digit (instance) distinguishes different situations with the same completion/category.

We define these specific reply codes:

- 200** Successful completion. This is the normal reply code when a command succeeds.

- 310** Message not found. The client asked for a specific message which is no longer available (it has been delivered to another client, or has been purged from the destination for some other reason).
- 311** Message is too large. The client asked to send a message which is larger than the server can accept at the present time. The client may retry at a later time.
- 320** Client already active. The server may restrict connections to one per named client (to ensure that subscriptions are unambiguous).
- 401** Service type not valid. The client asked for a service type that the server does not recognise or support.
- 402** Virtual path not valid. The client tried to open a connection to a virtual path that is not supported by the server.
- 403** Access to destination refused. The client asked to work with a destination but had insufficient authority to use it.
- 404** Destination not found. The client asked to work with a destination that does not exist on the server.
- 501** Framing error. The client sent a frame which the server could not decode.
- 502** Syntax error in arguments. The client sent invalid values for one or more fields. The reply text contains the name of the field which was invalid.
- 503** Command not valid at this time. The client sent a command that was illegal at the current time. This implies a programming error in the client.
- 504** Channel not ready. The client tried to work with a channel that was either undefined or closed. This implies a programming error in the client.
- 505** Handle not ready. The client tried to work with a handle that was either undefined or closed. This implies a programming error in the client.
- 506** Insufficient resources. The client tried to open more channels or handles than the server could support.
- 540** Functionality not implemented. The client tried to use some function which this version of the server does not implement.
- 541** Internal error detected. The server or client issuing the response needs intervention.
- 542** Message contents invalid. The server or client detected a corrupted or invalid message contents.

Clients should be designed to accept any reply code and handle it depending on the completion digit. E.g. clients should not test for a reply code like 200 but all reply codes 2xx.

3.4 Alternatives

We do not propose any alternatives at this moment.

3.5 Security Considerations

This proposal does not have any specific security considerations.

4 Comments on this Document

Comments by readers; these comments may be edited, incorporated, or removed by the author(s) of the document at any time.

4.1 Date, name

No comments at present.