GLOBAL MOBILE MESSAGING

Technical Specification

Q-CASTER 2.5

Tuesday, October 21, 2003

custsupport@quios.net

Table of Contents

1 Introduction

Authorized users of the Q-Caster services can programmatically submit XML messages through a simple HTTP interface. These messages are converted to SMS and distributed to mobile Handsets worldwide. A single message can be sent to many Handsets, or many messages can be sent to many Handsets. Messages are sent in real-time, and the sender receives real-time indication of success or error for each message. Q-Caster is ideal for sending large numbers of messages directly from a database or other content provider application.

Q-Caster 2.5 represents an improvement in functionality and feature set of the Q-Caster system. This version offers support for 8-bit messages, ringtones, flash messages, and dynamic originators.

The enhanced features of Q-Caster 2.5 will also appear in Q-Caster 3.0, in conjunction with an expanded feature set in 3.0.

1.1 Prerequisites

To submit messages to Q-Caster for transmission, the following conditions must be met:

- Provider must have a current, valid Contract for Message Distribution Services with Quios.
- Provider must have an existing Quios account with valid authentication information (username and password).
- Provider must have a Calling Application capable of transmitting Submissions in accordance with these specifications.
- Provider must have a valid access number for each Handset intended as a Message destination, including necessary country codes and area codes.
- At least one valid static IP number or range (CIDR) must be associated with Provider's username/password.

This technical specifications document assumes familiarity with the following standards, protocols, specifications, and/or RFCs:

- RFC 1945 “Hypertext Transfer Protocol – HTTP/1.0” or RFC 2616 “Hypertext Transfer Protocol – HTTP/1.1”
- ETSI GSM 03.38
- Internet-Draft “SSL Protocol Version 3.0 Specification”
- XML DTD syntax as specified in the W3C Recommendation “Extensible Markup Language (XML) 1.0 (Second Edition)”
- RFC 1738 “Uniform Resource Locators (URL)”
- Nokia Smart Messaging Spec
- Nokia Smart Messaging FAQ

1.2 System elements and terminology

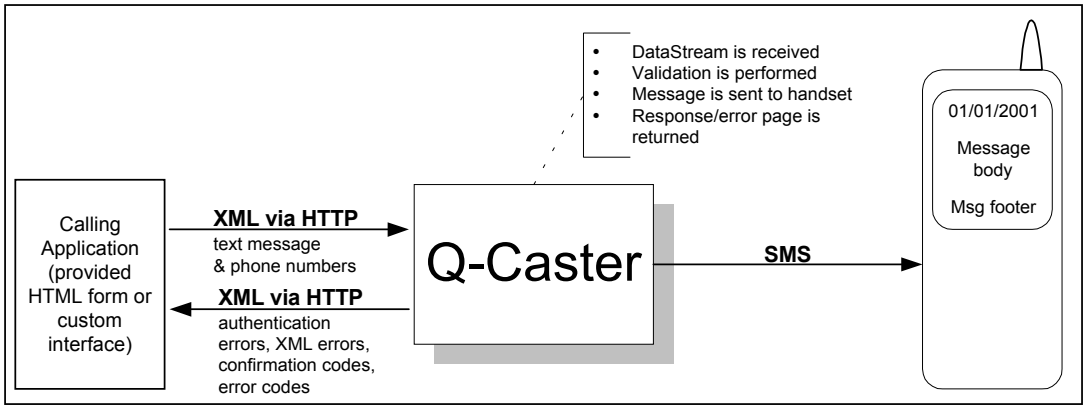
In addition to industry-standard terminology, this document defines additional terms as listed in Table 1-1.

Table 1-1 Terminology used in Q-Caster Technical Specifications

Calling Application	The programmatic interface that produces the Provider's message and receives resultant success/error notifications. The provided HTML testing interface is an example of a Calling Application; see Section 2.2 for more information.
Submission	The complete XML message transmitted to Q-Caster via HTTP POST by Provider.
Message	The complete text to be shown on the Handset display.
Handset	Mobile telephone, text pager, or other device capable of receiving SMS messages.
Provider	The organization that provides the Submission and sends it to Q-Caster for distribution to the Handsets.

Figure 1.1 indicates the flow of information between elements of the Q-Caster system.

Figure 1.1: Overview diagram of Q-Caster messaging



2 Technical Specifications

2.1 Connecting to the Q-Caster server

In order to transmit the Submission to the Q-Caster system for distribution, the Calling Application must first establish a network connection with the Q-Caster access server. The server connection URL is:

`http://qcaster.ewingz.com/qc25a/`

It is recommended that the connection always be made to the symbolic DNS name for the access server rather than to the IP address. The IP address associated with the server name is subject to change without notice.

2.2 Transmitting Submissions to Q-Caster

Submissions are transmitted to Q-Caster in an HTTP POST request. The interface for composing the request is the responsibility of the Calling Application. Submissions sent to the access server must conform to Sections 2.2.1 and 2.2.2, as well as either RFC 1945 or RFC 2616.

Quios provides a conformant HTML Calling Application for testing purposes. The server connection URL provided in Section 2.1 demonstrates the structure of a HTML Calling Application, in addition to serving as a direct connection to the Q-Caster service. When data is sent directly to this URL, the data enters the Q-Caster system. However, if a web browser connects to this URL but no data is posted, then the URL displays a conformant Calling Application that is available for use directly or as a model for development.

2.2.1 HTTP sequence

After establishing a network connection to the Q-Caster access server (see Section 2.1), the Calling Application transmits the HTTP POST request.

The request URI for the HTTP POST request is: `/qc25a/`

The request must contain the following HTTP headers:

- Host: `qcaster.ewingz.com`
- Content-type: application/x-www-form-urlencoded
- Content-length: *xxx*

(Replace *xxx* with the length of the HTTP POST body, as specified in RFC 1945 or RFC 2616.) Additional headers included in the HTTP POST request are discarded and do not adversely affect system function.

The HTTP POST request body must consist of a single attribute named *data*, whose value is an XML document that conforms to Section □.

After sending the request, the Calling Application does not close the network connection, but instead leaves it open in order to receive a response from the access server (see Section 2.3).

IMPORTANT: To reiterate, the HTTP POST request must contain these two headers:

- Content-type: application/x-www-form-urlencoded
- data=<conforming XML document>

2.2.2 XML Document

The data attribute of the HTTP POST request must contain a <Q-Caster> XML document that conforms to the specifications in this section. The attribute value may not begin with leading spaces, linefeed characters, or quotation marks. The current DTD for validating the Q-Caster XML document is contained in Listing 2-1; this DTD is subject to change without notice. An ASCII example of a conformant XML document is contained in Listing 2-2; actual submissions are encoded as specified in Section 2.2.1. Line breaks and spacing in Listing 2-2 are provided only as a convenience to the reader. Refer to Table 2-1 for information on the meanings of the input elements and attributes.

Listing 2-1 Q-Caster input DTD 1.0

```
<?xml version="1.0"?>
<!-- DTD Q-Caster Input v1.0 Copyright 2000-2002 eWingz
Systems Inc. -->
<!ELEMENT Q-Caster (message+)>
<!ATTLIST Q-Caster
  username CDATA #REQUIRED
  password CDATA #REQUIRED
  test_mode (on | off) "off"
>
<!ELEMENT message (device+)>
<!ATTLIST message
  text CDATA #REQUIRED
  originator CDATA #IMPLIED
>
<!ELEMENT device EMPTY>
<!ATTLIST device
  number CDATA #REQUIRED
>
```

Listing 2-2 Example Q-Caster input document

```
<?xml version="1.0"?>
<Q-Caster username="xxxxxxxxx" password="xxxxxxxxx"
 test_mode="off">
  <message text="This is one Text Message">
    <device number="11111111111"/>
  </message>
  <message text="This is another Text Message"
           originator="CompanyName">
    <device number="11111111111"></device>
    <device number="11111111111"/>
  </message>
</Q-Caster>
```

Table 2-1 Meaning of input elements and attributes

Elem/Attrib name	Meaning
username	Provider's Quios username for authentication.
password	Provider's Quios password for authentication.
test_mode	Attribute specifying whether to execute delivery of message. When test_mode is on, all authentication, validation, and parsing steps are performed, but the Message is not be sent to the Handsets and the Provider's account is not debited for Messages contained in this Submission.
message	Contains text and device information.
text	Message Body to transmit to all devices contained in the message element. May contain binary content, RTTTL content, or GSM 03.38 content. Limited to 4000 characters total.
originator	Sets the SMS originator to the specified string dynamically. Defaults to the originator string that is associated with this Provider. See Section 2.2.7 for details
device	Contains number and carrier information. The total number of device elements in a single Submission cannot exceed 100.
number	Numeric access number for destination Handset, including relevant country codes, area codes, and city codes. Any spaces or non-numeric characters are stripped from the input before it is parsed, and do not affect the number length or value. Number fields must be between 6 and 14 digits inclusive.

After sending the HTTP POST request, the Calling Application waits for a response from the access server.

2.2.3 Text messages

Characters intended for text display in the Message are encoded by the ETSI GSM 03.38 default alphabet and are limited to the characters available in that character set. A chart of the encodings and available characters are shown in Table 2-2; the complete list of encodings (and their Unicode encodings for reference) is in Appendix A.

The complete contents of the `text` attribute for a text message must match this regular expression:

```
^[\x00-\x7F]+
```

Different character sets are supported by different types of Handset hardware and mobile service carriers. Because Q-Caster cannot determine the specific character transmission and display capabilities for any particular Handset, text Messages are not guaranteed to display correctly. However, many Handsets and mobile service carriers, particularly those used outside the United States, are capable of transmitting and displaying ETSI GSM 03.38 characters. In most cases these characters transmit and display correctly.

Table 2-2 Characters used by ETSI GSM 03.38 default alphabet

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	@	£	$	¥	è	é	ù	ì	ò	ç	LF	Ø	ø	CR	Å	å
10	Δ	_	Φ	Γ	Λ	Ω	Π	Ψ	Σ	Θ	Ξ	esc	Æ	æ	ß	É
20	spc	!	"	#	¤	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	¡	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	Ä	Ö	Ñ	Ü	§
60	¿	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	ä	ö	ñ	ü	à

Note: See Appendix A for an explanation of the characters marked with □ and of some variations in the display of certain characters. For example, the Euro currency symbol is part of the extended characters available by using the 0x1B escape sequence; use &#x1B;e to display the Euro symbol.

Characters sent as part of the text attribute are representations of the characters (letters and digits) that will appear on the handset. However, in several cases these characters cannot use their ordinary representation: when they are not part of the ASCII character set, when they are reserved characters for one of the protocols used in transmission (XML, URL, etc.), when they are non-printable sequences, and so on. Each character can be represented by a sequence that includes its hexadecimal value that can be considered equivalent to the character itself; this sequence is its encoding.

Characters in the text attribute must be encoded in three ways: first, the correct ETSI GSM 03.38 character code must be used to properly represent non-ASCII characters, then any special characters in XML must have proper entity encoding, and finally the text must be URL encoded for proper transmission to Q-Caster.

2.2.3.1 Encoding non-ASCII characters

In Table 2-2, shading indicates the characters that use the same hexadecimal code in both ETSI GSM 03.38 and ASCII (ISO-8859-1). These shaded characters can be transmitted to Q-Caster as ASCII characters because they will be interpreted correctly. Characters that are not shaded must be transmitted using their ETSI GSM 03.38 hexadecimal encodings. For example, the character 'a' may remain 'a', but the character '@' must be represented by the XML encoding '&#x00;'.

2.2.3.2 XML entity encoding

Because certain characters have a special meaning in XML, if those characters are present in the text attribute they must be replaced by their entity encodings prior to submission. Alternately, they can be replaced by their hexadecimal Unicode encodings; this solution is equivalent. The special characters and their encodings are shown below.

Unencoded	Entity encoding	Hexadecimal Unicode encoding
&	%26amp%3B	%26#x26%3B
>	%26gt%3B	%26#x3E%3B
<	%26lt%3B	%26#x3C%3B
'	%26apos%3B	%26#x27%3B
"	%26quot%3B	%26#x22%3B

2.2.3.3 URL encoding

Finally, any characters with special meaning within the URL scheme must be encoded. Fortunately, many of these characters have already been encoded because they are non-ASCII or because they require XML encoding. The remaining characters must be URL encoded before sending; these characters are shown below.

Unencoded	URL encoding
%	%25
+	%2B
#	%23
/	%2F
:	%3A
;	%3B
=	%3D
?	%3F

More information on URL encoding is available in RFC 1738 “Uniform Resource Locators (URL)”. Proper URL encoding (conforming with RFC 1738) is the responsibility of the Calling Application and any libraries that it uses for submitting the POST request. Many types of Calling Application (such as a web browsers) perform the URL encoding automatically.

Note: Using this additional URL encoding in an XML submission is technically an error in the XML. The XML document is implicitly declared to be encoded as UTF-8, but these additional URL encodings are not valid in UTF-8. However, this is not a fatal error, and will cause no difficulties in using the Q-Caster system.

Note: Especially when using Microsoft's MSXML/HTTP component, check the Calling Application's capabilities carefully to ensure that character encoding is performed correctly.

2.2.3.4 Encoding examples

Here is an example of how to encode a message. Consider a message that contains an email address: `custsupport@quios.net`.

The @ character is problematic, because unlike the other characters in the example, its ETSI GSM 03.38 encoding (&#x00;) differs from its ASCII encoding. Therefore, that character must be represented by an encoding sequence: `custsupport&#x00;quios.net`.

The next encoding requirement is XML entity encoding. Because this string contains no XMLspecial characters, no XML encoding is required.

Finally, the string must be URL encoded. The &, #, and ; characters require URL encoding. Therefore, the final representation of the string is: `custsupport%26%23x00%3Bquios.net`. This is the value of the text attribute of this submission.

For testing purposes, the following example submission is XML encoded and URL encoded, and displays the entire ETSI GSM 03.38 default alphabet on the handset.

```
<?xml version="1.0"?>
<Q-Caster username='xxxx' password='xxxx' test_mode='off'>
  <message
text='%26%23x00%3B%26%23x01%3B%26%23x02%3B%26%23x03%3B%26%23x0
4%3B%26%23x05%3B%26%23x06%3B%26%23x07%3B%26%23x08%3B%26%23x09%
3B%26%23x0A%3B%26%23x0B%3B%26%23x0C%3B%26%23x0D%3B%26%23x0E%3B
%26%23x0F%3B%26%23x10%3B%26%23x11%3B%26%23x12%3B%26%23x13%3B%2
6%23x14%3B%26%23x15%3B%26%23x16%3B%26%23x17%3B%26%23x18%3B%26%
23x19%3B%26%23x1A%3B%26%23x1B%3B%26%23x1C%3B%26%23x1D%3B%26%23
x1E%3B%26%23x1F%3B%26%23x20%3B%26%23x21%3B%26%23x22%3B%26%23x2
3%3B%26%23x24%3B%26%23x25%3B%26%23x26%3B%26%23x27%3B%26%23x28%
3B%26%23x29%3B%26%23x2A%3B%26%23x2B%3B%26%23x2C%3B%26%23x2D%3B
%26%23x2E%3B%26%23x2F%3B%26%23x30%3B%26%23x31%3B%26%23x32%3B%2
6%23x33%3B%26%23x34%3B%26%23x35%3B%26%23x36%3B%26%23x37%3B%26%
23x38%3B%26%23x39%3B%26%23x3A%3B%26%23x3B%3B%26%23x3C%3B%26%23
x3D%3B%26%23x3E%3B%26%23x3F%3B%26%23x40%3B%26%23x41%3B%26%23x4
2%3B%26%23x43%3B%26%23x44%3B%26%23x45%3B%26%23x46%3B%26%23x47%
3B%26%23x48%3B%26%23x49%3B%26%23x4A%3B%26%23x4B%3B%26%23x4C%3B
%26%23x4D%3B%26%23x4E%3B%26%23x4F%3B%26%23x50%3B%26%23x51%3B%2
6%23x52%3B%26%23x53%3B%26%23x54%3B%26%23x55%3B%26%23x56%3B%26%
23x57%3B%26%23x58%3B%26%23x59%3B%26%23x5A%3B%26%23x5B%3B%26%23
x5C%3B%26%23x5D%3B%26%23x5E%3B%26%23x5F%3B%26%23x60%3B%26%23x6
1%3B%26%23x62%3B%26%23x63%3B%26%23x64%3B%26%23x65%3B%26%23x66%
3B%26%23x67%3B%26%23x68%3B%26%23x69%3B%26%23x6A%3B%26%23x6B%3B
%26%23x6C%3B%26%23x6D%3B%26%23x6E%3B%26%23x6F%3B%26%23x70%3B%2
6%23x71%3B%26%23x72%3B%26%23x73%3B%26%23x74%3B%26%23x75%3B%26%
23x76%3B%26%23x77%3B%26%23x78%3B%26%23x79%3B%26%23x7A%3B%26%23
x7B%3B%26%23x7C%3B%26%23x7D%3B%26%23x7E%3B%26%23x7F%3B'>
    <device number='11111111111' />
  </message>
</Q-Caster>
```

2.2.3.5 Line breaks in SMS messages

As mentioned previously, the display of characters on handsets can vary between manufacturers and phone models. However, many phones will end a line and begin the next line in response to this ETSI GSM 03.38 character code: `
`

2.2.3.6 Longer messages

Text messages of more than 160 characters are split into multiple SMS transmissions of 160 characters each. If the message must be split in a

particular place (between words, for example), then the calling application must submit it as multiple messages, split in the proper place.

2.2.3.7 Special rules for the first character

If the first character of the message is a single "\", then the message will be sent as a flash message.

The ETSI GSM 03.38 character 0x5C (Ö) cannot begin a non-flash message; it is interpreted as ASCII character 0x5C (\) during part of the submission process, and so will cause the message to be sent as a flash message.

If the first two characters in the message are two "\" characters, then the message will not be sent as flash, and the message will begin with the Ö character displayed on the handset. See Section 2.2.4 for more details on flash messages.

2.2.3.8 RTTTL content

Text messages are inspected for RTTTL content. If the text message is determined to be an RTTTL message, then it is transmitted to the handset as a ringtone.

2.2.3.9 Timestamps

Quios uses a number of geographically diverse SMSCs. Due to the nature of SMSC platforms, each SMSC normally timestamps each message in its local timezone. In some cases, the SMSC is located in a different timezone from the destination handset.

Quios cannot provide a timestamp that is guaranteed to be accurate for the location of the destination handset. Applications that require accurate local timestamps should include the timestamp in the message text.

2.2.4 Flash messages

Text messages can be sent as flash SMS messages (class 0 messages). These messages appear on the handset immediately, and are not stored to memory. Flash capabilities are dependent on the handset's manufacturer and model; results on hardware that is not flash-capable are undefined.

To send a flash message, prepend the "\" character to the text message. The complete contents of the `text` attribute must match this regular expression:

```
^\\[\x00-\x7F]+
```

Text messages of more than 160 characters (including the prepended "\") are split into multiple SMS transmissions of 160 characters each. The first message is sent as flash SMS, and all subsequent messages are sent as standard SMS.

Flash messages must adhere to all specifications for text messages. See Section 2.2.3 for these specifications.

2.2.5 RTTTL messages

Instead of a text message for display on the handset, the Message can consist of an RTTTL ringtone for transmission to the handset. Q-Caster 2.5 supports transmission of RTTTL. Lengthy ringtones are automatically split

into multiple concatenated messages and need not be split before transmission to Q-Caster.

Each incoming message to Q-Caster is inspected for RTTTL content. If RTTTL is detected in the `text` attribute, then the message is transmitted as a ringtone rather than as a text message.

Note: Certain types of errors in the RTTTL content causes the message to be identified as a text message, not an RTTTL message. For example, using a hyphen "-" character in a song title is not valid in RTTTL, and will cause Q-Caster to identify and deliver the message as a series of SMS text messages. Inaddition, using an invalid tempo (such as tempo=3) causes the message to fail identification as an RTTTL submission.

The RTTTL content is parsed for errors, and if encoding problems are encountered, an error `err_general_failure` is reported to the calling application. If no errors are detected, then the RTTTL message is sent to the handset as a ringtone.

The calling application can set the RTTTL attributes for the message. The <name> attribute is required; behavior of RTTTL messages without a <name> attribute is unspecified. Other attributes are optional. For the following attributes, default values will be provided when the calling application omits values for these attributes:

Attribute name	Default value
duration	4
scale	6
beats	63
looping	0

The complete specification for RTTTL is in Appendix B.

RTTTL capabilities are dependent on the handset's manufacturer and model; results on hardware that is not RTTTL-capable are undefined. RTTTL ringtone support is available only on certain Nokia hardware (models 3210, 3310, 6110, 6130, 6150, 6210, 6250, 7110, 8110i, 8210, 8810, 8850, 8890, 9110, 9110i, 9210). This information is subject to change.

The RTTTL feature is under active development. Please report any issues to custsupport@quios.net.

2.2.6 8-bit binary messages

Q-Caster 2.5 accepts messages in 8-bit binary format. It performs no validation or encoding; the calling application is responsible for proper validation and encoding.

The message can consist of a user data header, a space " " character, and a binary body. This type of message is transmitted with UDHI indicated. The user data header and binary body are limited to 280 characters, with the space character bringing the total to 281. The SMS header is not included in this length limitation.

Each binary SMS is limited to a maximum of 140 encoded bytes and 281 total characters. However, for larger binary Messages, the Calling Application must concatenate multiple SMSs as a single Message submission, using a colon character ":" as a divider between SMSs. The Quios system splits these messages at the colon and delivers each SMS separately. The SMSs are reassembled upon reaching the Handset. Multipart Messages that are sent as separate Message submissions risk reaching the Handset in an incorrect order, or failing reassembly upon arrival at the Handset.

The regular expression for all acceptable binary messages in the `text` attribute is:

```
^(?:(?:[0-9A-F][0-9A-F])+ (?:[0-9A-F][0-9A-F])+:?)+$
```

Binary capabilities are dependent on the handset's manufacturer and model; results on hardware that is not smart message-capable are undefined.

Note: Although Quios cannot provide detailed support regarding binary content, online resources are available to assist in assembling binary content, particularly for Nokia Smart Messaging. Refer to the Nokia Smart Messaging Spec or the Nokia Smart Messaging FAQ for assistance with Smart Messaging formats.

Both of these documents are available, after registering, from the Nokia website http://www.forum.nokia.com/. For the Smart Messaging Spec, choose Technologies, then Messaging, then Documents. For the Smart Messaging FAQ, choose Technologies, then Messaging, then Smart Messaging.

Note: The Calling Application is responsible for setting the MCC/MCN octets of operator logos.

2.2.7 Message originators

The requirements for the originator field depend on the Provider's contractual agreement with Quios. Quios allows different types of originators. These specifications for originators apply to both the default originator field and to dynamic originator settings.

A standard originator must contain at least one non-numeric character, must be at least 2 and no more than 11 alphanumeric characters, and must match the following regular expression:

```
/^[-.+0-9A-Za-z]{2,11}$/
```

With prior contractual agreement, a Provider can use an International originator. With this type of originator, the Handset can call directly back to the message sender as represented in the originator. This type of originator must use a valid international format MSISDN in the originator field, optionally preceded by the plus "+" character. International originators must be at least 6 and no more than 15 digits. A leading plus "+" character forces the number to be interpreted as an international number.

With prior contractual agreement, a Provider can use a National originator (also called a Short Code). This type of originator must at least one and no more than 5 digits, and may not include a plus "+" character. A leading minus "-" character forces the number to be interpreted as a local number.

2.3 Receiving responses from Q-Caster

Upon submission to Q-Caster, the Submission undergoes a series of validity checks in the following order: XML parsing, authentication, and number/message evaluation. The Calling Application receives a single response, described in Sections 2.3.1 through 2.3.3, based on the results of these validity checks. Although the validity checks occur in the order described, responses are returned in an indeterminate order depending on the severity and nature of the error. After transmitting the response, the access server closes the network connection.

A response consisting of an HTTP 500 Internal Server Error indicates that the Quios webserver is currently unavailable.

Note: A return_code="ok" response does not indicate successful delivery to the Handset. This preliminary status indicates only that Q-Caster has determined that the request appears to have a valid device number and originator, and that the contents seem to be encoded in a valid form. This response confirms that the message has entered the Q-Caster system for delivery processing, but does not indicate delivery status. The Q-Caster 2.5a interface does not provide delivery notification.

Responses from the access server conform to the DTD shown in Listing 2-3.

Listing 2-3 Q-Caster output DTD 1.0

```
<?xml version="1.0"?>
<!-- DTD Q-Caster Response v1.0 Copyright 2000-2002 eWingz
Systems Inc. -->
<!ELEMENT Q-Caster (message*, fatal_errors?)>
<!ELEMENT fatal_errors (#PCDATA)>
<!ELEMENT message (device+)>
<!ATTLIST message
  text CDATA #IMPLIED
>
<!ELEMENT device (errors*)>
<!ATTLIST device
  return_code (ok | failed) #REQUIRED
  number CDATA #REQUIRED
>
<!ELEMENT errors (#PCDATA)>
```

2.3.1 Response to timeout

Submissions containing more than 100 device elements are subject to timeout by the access server. In the case of server timeouts, no response is returned to the Calling Application, no SMS messages are transmitted to Handsets, and the Provider account is not debited. The network connection might be closed by the access server; this behavior is undefined and depends upon many variables. The Calling Application may optionally test the network connection while awaiting server response. No explicit indication of success or failure is sent to the Calling Application.

2.3.2 Response to XML parsing failure

If the server responds before timing out, the XML is parsed and validated against the rules given in Section 2.22. In the event of XML parsing failure,

the entire Submission is rejected, no SMS messages are transmitted to Handsets, and the Provider account is not debited.

XML parsing failure is indicated by the return document's fatal_errors element containing the code err_bad_xml. An example of the access server response to an XML parsing failure is shown in Listing 2-4.

Listing 2-4 Example of XML parsing failure

```
<?xml version="1.0"?>
<Q-Caster>
<fatal_errors>err_bad_xml</fatal_errors>
</Q-Caster>
```

XML parsing success results in the Submission being passed to the authentication module. No explicit indication of success is sent to the Calling Application.

2.3.3 Response to authentication failure

If XML parsing is successful, username/password authentication is performed. The username, password, and URL are case-sensitive. Authentication failure results from any of the following conditions:

- Username field is blank
- Username attribute is missing
- Password attribute is missing
- Password given does not authenticate username given

In the event of authentication failure, the entire Submission is rejected, no SMS messages are transmitted to Handsets, and the Provider account is not debited.

Authentication failure is indicated by an HTTP 401 (unauthorized) error. An HTML document is returned, containing information that helps to explain the problem.

Authentication success results in the Submission being passed to the message parser. No explicit indication of success is sent to the Calling Application.

2.3.4 Response to message success/failure

If authentication and parsing are successful, then Q-Caster evaluates each number and message in the XML document and issues an XML document in response to the Calling Application. A single response is returned, containing success or failure information for each device indicated in the Submission. The Provider account is debited for the successful messages, which are transmitted to Handsets immediately after being evaluated. After transmitting the response, the access server closes the network connection to the Calling Application.

Responses conform to the DTD shown in Listing 2-3. An example of a valid response is contained in Listing 2-6.

Listing 2-6 Example Q-Caster output document

```
<?xml version="1.0"?>
<Q-Caster>
  <message text="   ">
    <device number="11111111111" return_code="failed">
      <errors>err_missing_message</errors>
    </device>
    <device number="" return_code="failed">
      <errors>err_missing_number</errors>
      <errors>err_checknumber_failed</errors>
    </device>
    <device number="12345678901" return_code="ok" />
    <device number="1" return_code="failed">
      <errors>err_number_wrong_length</errors>
    </device>
    <device number="12222222222" return_code="failed">
      <errors>err_blocked_address</errors>
    </device>
  </message>
  <message text="">
    <device number="3211111111" return_code="failed">
      <errors>err_missing_message</errors>
    </device>
  </message>
</Q-Caster>
```

Successful delivery of the SMS message to a Handset is indicated by the return_code attribute on the affected device holding the value "ok". Failed delivery is indicated by the return_code attribute holding the value "failed".

In the event of a failed delivery, the affected device contains one or more error elements. Each error element contains a code indicating the reason for the failure. After resolving the errors, the Calling Application can resubmit failed devices, provided that such resubmissions conform to this entire specification. An explanation of the error codes is contained in Table 2-3.

Note: The SMS is delivered to the carrier that administrates the destination Handset. If the Handset is roaming, it is the responsibility of the carrier to forward the message to the appropriate service.

Table 2-3 Response error codes and their meanings

Error code	Meaning
err_bad_xml	The XML message cannot be parsed, or does not conform to the published DTD.
err_blocked_address	Messages cannot be delivered to this Handset. This error is accompanied by the two detailed error messages below.
err_blocked_by_account	Accompanies err_blocked_address, and is used when the handset is blocked from messages that originate from this Provider.
err_blocked_globally	Accompanies err_blocked_address, and is used when the handset is blocked from messages that originate from any Provider.
err_checknumber_failed	The country code could not be determined for this number.
err_general_failure	A generic error message used when a more specific error indication is not available.
err_invalid_content	The message contains invalid content. For example, this error results from messages containing characters that are outside the range of 0x00-0x7F.
err_message_limit	Provider has exceeded the authorized number of messages for a particular time period. This error is accompanied by the two detailed error messages below.
err_message_limit_daily	Accompanies err_message_limit, and indicates that Provider's daily message limit has been exceeded.
err_message_limit_monthly	Accompanies err_message_limit, and indicates that Provider's monthly message limit has been exceeded.
err_missing_carrier	Obsolete.
err_missing_message	<message text=“”> attribute contains only blank characters.
err_missing_number	<number> element contains only blank characters .
err_number_wrong_length	Number of digits in Handset number is not correct for the provided operator code and prefix.
err_unknown_ip_address	Indicates that the incoming message is not from a recognized IP address. Q-Caster will accept only messages that originate from IP addresses that are registered to the given Provider.

3 Appendix A: Available GSM Characters and Their Encodings

http://www.unicode.org/Public/MAPPINGS/ETSI/GSM0338.TXT

```
#
#       Name:             GSM 03.38 to Unicode
#       Unicode version:  3.0
#       Table version:    1.1
#       Table format:     Format A
#       Date:             2000 May 30
#       Authors:          Ken Whistler <kenw@sybase.com>,
#                         Kent Karlsson <keka@im.se>,
#                         Markus Kuhn <mkuhn@acm.org>
#
#       Copyright (c) 2000 Unicode, Inc.  All Rights reserved.
#
#       This file is provided as-is by Unicode, Inc. (The Unicode Consortium).
#       No claims are made as to fitness for any particular purpose.  No
#       warranties of any kind are expressed or implied.  The recipient
#       agrees to determine applicability of information provided.  If this
#       file has been provided on optical media by Unicode, Inc., the sole
#       remedy for any claim will be exchange of defective media within 90
#       days of receipt.
#
#       Unicode, Inc. hereby grants the right to freely use the information
#       supplied in this file in the creation of products supporting the
#       Unicode Standard, and to make copies of this file in any form for
#       internal or external distribution as long as this notice remains
#       attached.
#
#       General notes:
#
#       This table contains the data the Unicode Consortium has on how
#       ETSI GSM 03.38 7-bit default alphabet characters map into Unicode.
#       This mapping is based on ETSI TS 100 900 V7.2.0 (1999-07), with
#       a correction of 0x09 to *small* c-cedilla, instead of *capital*
#       C-cedilla.
#
#       Format:  Three tab-separated columns
#                Column #1 is the ETSI GSM 03.38 7-bit default alphabet
#                             code (in hex as 0xXX, or 0xXXXX for double-byte
#                             sequences)
#                Column #2 is the Unicode scalar value (in hex as 0xXXXX)
#                Column #3 the Unicode name (follows a comment sign, '#')
#
#       The entries are in ETSI GSM 03.38 7-bit default alphabet code order.
#
#       Note that ETSI GSM 03.38 also allows for the use of UCS-2 (UTF-16
#       restricted to the BMP) in GSM/SMS messages.
#
#       Note also that there are commented Greek mappings for some
#       capital Latin characters. This follows from the clear intent
#       of the ETSI GSM 03.38 to have glyph coverage for the uppercase
#       Greek alphabet by reusing Latin letters that have the same
#       form as an uppercase Greek letter. Conversion implementations
#       should be aware of this fact.
#
#       The ETSI GSM 03.38 specification shows an uppercase C-cedilla
#       glyph at 0x09. This may be the result of limited display
#       capabilities for handling characters with descenders. However, the
#       language coverage intent is clearly for the lowercase c-cedilla, as shown
#       in the mapping below. The mapping for uppercase C-cedilla is shown
```

```
#       in a commented line in the mapping table.
#
#       The ESC character 0x1B is
#       mapped to the no-break space character, unless it is part of a
#       valid ESC sequence, to facilitate round-trip compatibility in
#       the presence of unknown ESC sequences.
#
#       0x00 is NULL (when followed only by 0x00 up to the
#       end of (fixed byte length) message, possibly also up to
#       FORM FEED.  But 0x00 is also the code for COMMERCIAL AT
#       when some other character (CARRIAGE RETURN if nothing else)
#       comes after the 0x00.
#
#       Version history
#       1.0 version: first creation
#       1.1 version: fixed problem with the wrong line being a comment,
#                       added text regarding 0x00's interpretation,
#                       added second mapping for C-cedilla,
#                       added mapping of 0x1B escape to NBSP for display.
#
#       Updated versions of this file may be found in:
#               <ftp://ftp.unicode.org/Public/MAPPINGS/>
#
#       Any comments or problems, contact <errata@unicode.org>
#       Please note that <errata@unicode.org> is an archival address;
#       notices will be checked, but do not expect an immediate response.
#
0x00    0x0040  #       COMMERCIAL AT
#0x00   0x0000  #       NULL (see note above)
0x01    0x00A3  #       POUND SIGN
0x02    0x0024  #       DOLLAR SIGN
0x03    0x00A5  #       YEN SIGN
0x04    0x00E8  #       LATIN SMALL LETTER E WITH GRAVE
0x05    0x00E9  #       LATIN SMALL LETTER E WITH ACUTE
0x06    0x00F9  #       LATIN SMALL LETTER U WITH GRAVE
0x07    0x00EC  #       LATIN SMALL LETTER I WITH GRAVE
0x08    0x00F2  #       LATIN SMALL LETTER O WITH GRAVE
0x09    0x00E7  #       LATIN SMALL LETTER C WITH CEDILLA
#0x09   0x00C7  #       LATIN CAPITAL LETTER C WITH CEDILLA (see note above)
0x0A    0x000A  #       LINE FEED
0x0B    0x00D8  #       LATIN CAPITAL LETTER O WITH STROKE
0x0C    0x00F8  #       LATIN SMALL LETTER O WITH STROKE
0x0D    0x000D  #       CARRIAGE RETURN
0x0E    0x00C5  #       LATIN CAPITAL LETTER A WITH RING ABOVE
0x0F    0x00E5  #       LATIN SMALL LETTER A WITH RING ABOVE
0x10    0x0394  #       GREEK CAPITAL LETTER DELTA
0x11    0x005F  #       LOW LINE
0x12    0x03A6  #       GREEK CAPITAL LETTER PHI
0x13    0x0393  #       GREEK CAPITAL LETTER GAMMA
0x14    0x039B  #       GREEK CAPITAL LETTER LAMDA
0x15    0x03A9  #       GREEK CAPITAL LETTER OMEGA
0x16    0x03A0  #       GREEK CAPITAL LETTER PI
0x17    0x03A8  #       GREEK CAPITAL LETTER PSI
0x18    0x03A3  #       GREEK CAPITAL LETTER SIGMA
0x19    0x0398  #       GREEK CAPITAL LETTER THETA
0x1A    0x039E  #       GREEK CAPITAL LETTER XI
0x1B    0x00A0  #       ESCAPE TO EXTENSION TABLE (or displayed as NBSP, see
note)
0x1B0A  0x000C  #       FORM FEED
0x1B14  0x005E  #       CIRCUMFLEX ACCENT
0x1B28  0x007B  #       LEFT CURLY BRACKET
0x1B29  0x007D  #       RIGHT CURLY BRACKET
0x1B2F  0x005C  #       REVERSE SOLIDUS
0x1B3C  0x005B  #       LEFT SQUARE BRACKET
0x1B3D  0x007E  #       TILDE
0x1B3E  0x005D  #       RIGHT SQUARE BRACKET
0x1B40  0x007C  #       VERTICAL LINE
0x1B65  0x20AC  #       EURO SIGN
```

```
0x1C    0x00C6  #       LATIN CAPITAL LETTER AE
0x1D    0x00E6  #       LATIN SMALL LETTER AE
0x1E    0x00DF  #       LATIN SMALL LETTER SHARP S (German)
0x1F    0x00C9  #       LATIN CAPITAL LETTER E WITH ACUTE
0x20    0x0020  #       SPACE
0x21    0x0021  #       EXCLAMATION MARK
0x22    0x0022  #       QUOTATION MARK
0x23    0x0023  #       NUMBER SIGN
0x24    0x00A4  #       CURRENCY SIGN
0x25    0x0025  #       PERCENT SIGN
0x26    0x0026  #       AMPERSAND
0x27    0x0027  #       APOSTROPHE
0x28    0x0028  #       LEFT PARENTHESIS
0x29    0x0029  #       RIGHT PARENTHESIS
0x2A    0x002A  #       ASTERISK
0x2B    0x002B  #       PLUS SIGN
0x2C    0x002C  #       COMMA
0x2D    0x002D  #       HYPHEN-MINUS
0x2E    0x002E  #       FULL STOP
0x2F    0x002F  #       SOLIDUS
0x30    0x0030  #       DIGIT ZERO
0x31    0x0031  #       DIGIT ONE
0x32    0x0032  #       DIGIT TWO
0x33    0x0033  #       DIGIT THREE
0x34    0x0034  #       DIGIT FOUR
0x35    0x0035  #       DIGIT FIVE
0x36    0x0036  #       DIGIT SIX
0x37    0x0037  #       DIGIT SEVEN
0x38    0x0038  #       DIGIT EIGHT
0x39    0x0039  #       DIGIT NINE
0x3A    0x003A  #       COLON
0x3B    0x003B  #       SEMICOLON
0x3C    0x003C  #       LESS-THAN SIGN
0x3D    0x003D  #       EQUALS SIGN
0x3E    0x003E  #       GREATER-THAN SIGN
0x3F    0x003F  #       QUESTION MARK
0x40    0x00A1  #       INVERTED EXCLAMATION MARK
0x41    0x0041  #       LATIN CAPITAL LETTER A
#0x41   0x0391  #       GREEK CAPITAL LETTER ALPHA
0x42    0x0042  #       LATIN CAPITAL LETTER B
#0x42   0x0392  #       GREEK CAPITAL LETTER BETA
0x43    0x0043  #       LATIN CAPITAL LETTER C
0x44    0x0044  #       LATIN CAPITAL LETTER D
0x45    0x0045  #       LATIN CAPITAL LETTER E
#0x45   0x0395  #       GREEK CAPITAL LETTER EPSILON
0x46    0x0046  #       LATIN CAPITAL LETTER F
0x47    0x0047  #       LATIN CAPITAL LETTER G
0x48    0x0048  #       LATIN CAPITAL LETTER H
#0x48   0x0397  #       GREEK CAPITAL LETTER ETA
0x49    0x0049  #       LATIN CAPITAL LETTER I
#0x49   0x0399  #       GREEK CAPITAL LETTER IOTA
0x4A    0x004A  #       LATIN CAPITAL LETTER J
0x4B    0x004B  #       LATIN CAPITAL LETTER K
#0x4B   0x039A  #       GREEK CAPITAL LETTER KAPPA
0x4C    0x004C  #       LATIN CAPITAL LETTER L
0x4D    0x004D  #       LATIN CAPITAL LETTER M
#0x4D   0x039C  #       GREEK CAPITAL LETTER MU
0x4E    0x004E  #       LATIN CAPITAL LETTER N
#0x4E   0x039D  #       GREEK CAPITAL LETTER NU
0x4F    0x004F  #       LATIN CAPITAL LETTER O
#0x4F   0x039F  #       GREEK CAPITAL LETTER OMICRON
0x50    0x0050  #       LATIN CAPITAL LETTER P
#0x50   0x03A1  #       GREEK CAPITAL LETTER RHO
0x51    0x0051  #       LATIN CAPITAL LETTER Q
0x52    0x0052  #       LATIN CAPITAL LETTER R
0x53    0x0053  #       LATIN CAPITAL LETTER S
0x54    0x0054  #       LATIN CAPITAL LETTER T
#0x54   0x03A4  #       GREEK CAPITAL LETTER TAU
```

```
0x55    0x0055  #       LATIN CAPITAL LETTER U
#0x55   0x03A5  #       GREEK CAPITAL LETTER UPSILON
0x56    0x0056  #       LATIN CAPITAL LETTER V
0x57    0x0057  #       LATIN CAPITAL LETTER W
0x58    0x0058  #       LATIN CAPITAL LETTER X
#0x58   0x03A7  #       GREEK CAPITAL LETTER CHI
0x59    0x0059  #       LATIN CAPITAL LETTER Y
0x5A    0x005A  #       LATIN CAPITAL LETTER Z
#0x5A   0x0396  #       GREEK CAPITAL LETTER ZETA
0x5B    0x00C4  #       LATIN CAPITAL LETTER A WITH DIAERESIS
0x5C    0x00D6  #       LATIN CAPITAL LETTER O WITH DIAERESIS
0x5D    0x00D1  #       LATIN CAPITAL LETTER N WITH TILDE
0x5E    0x00DC  #       LATIN CAPITAL LETTER U WITH DIAERESIS
0x5F    0x00A7  #       SECTION SIGN
0x60    0x00BF  #       INVERTED QUESTION MARK
0x61    0x0061  #       LATIN SMALL LETTER A
0x62    0x0062  #       LATIN SMALL LETTER B
0x63    0x0063  #       LATIN SMALL LETTER C
0x64    0x0064  #       LATIN SMALL LETTER D
0x65    0x0065  #       LATIN SMALL LETTER E
0x66    0x0066  #       LATIN SMALL LETTER F
0x67    0x0067  #       LATIN SMALL LETTER G
0x68    0x0068  #       LATIN SMALL LETTER H
0x69    0x0069  #       LATIN SMALL LETTER I
0x6A    0x006A  #       LATIN SMALL LETTER J
0x6B    0x006B  #       LATIN SMALL LETTER K
0x6C    0x006C  #       LATIN SMALL LETTER L
0x6D    0x006D  #       LATIN SMALL LETTER M
0x6E    0x006E  #       LATIN SMALL LETTER N
0x6F    0x006F  #       LATIN SMALL LETTER O
0x70    0x0070  #       LATIN SMALL LETTER P
0x71    0x0071  #       LATIN SMALL LETTER Q
0x72    0x0072  #       LATIN SMALL LETTER R
0x73    0x0073  #       LATIN SMALL LETTER S
0x74    0x0074  #       LATIN SMALL LETTER T
0x75    0x0075  #       LATIN SMALL LETTER U
0x76    0x0076  #       LATIN SMALL LETTER V
0x77    0x0077  #       LATIN SMALL LETTER W
0x78    0x0078  #       LATIN SMALL LETTER X
0x79    0x0079  #       LATIN SMALL LETTER Y
0x7A    0x007A  #       LATIN SMALL LETTER Z
0x7B    0x00E4  #       LATIN SMALL LETTER A WITH DIAERESIS
0x7C    0x00F6  #       LATIN SMALL LETTER O WITH DIAERESIS
0x7D    0x00F1  #       LATIN SMALL LETTER N WITH TILDE
0x7E    0x00FC  #       LATIN SMALL LETTER U WITH DIAERESIS
0x7F    0x00E0  #       LATIN SMALL LETTER A WITH GRAVE
```

4 Appendix B: RTTTL Specification

The values listed here are acceptable; any other values can cause indeterminate results.

```
<ringing-tones-text-transfer-language> :=
<name> <sep> [<defaults>] <sep> <note-command>+

<name> := <char>+ ; maximum name length 10 characters

<sep> := ":"

<defaults> :=
<def-note-duration> := 'd'
<def-note-scale>    := 'o'
<def-beats>         := 'b'
<def-style>         := 's'
<def-looping>       := 'l'

If not specified, defaults are

4  = duration
6  = scale
63 = beats-per-minute

Valid in tone section: o, b, s

 <note-command> :=
[<duration>] <note> [<scale>] [<special-duration>] <delimiter>

<duration> :=
"1"  Full 1/1 note
"2"  1/2 note
"4"  1/4 note
"8"  1/8 note
"16"  1/16 note
"32" 1/32 note

<note> :=
"P"  pause
"C"
"C#"
"D"
"D#"
"E"
"F"
"F#"
"G"
"G#"
"A"
"A#"
"B"
"H"

<scale> :=
"4"  Note A is 440Hz
"5"  Note A is 880Hz
"6"  Note A is 1.76 kHz
"7"  Note A is 3.52 kHz

<special-duration> :=
"."  Dotted note
";"  Double dotted note
"&"  2/3 length

<delimiter> := ","

Acceptable values for B are:
25|28|31|35|40|45|50|56|63|70|80|90|100|112|125|140|160|180|
200|225|250|285|320|355|400|450|500|656|635|715|800|900

Acceptable values for Volume are: 1 through 15, inclusive
```

5 Appendix C: Document change log

Date	Section	Description of change
2003/10/21	Sec 2.2.7	Changed description of + and - behavior in originators.
2003/10/21	App B	Added list of acceptable values for parameters.
2003/05/20	Section 2.3	Added information on HTTP 500 Internal Server Error
2002/10/23	Table 2-3	Improved accuracy of description of err_checknumber_failed and err_number_wrong_length
2002/10/11	Section 2.2.7	Added information RE multipart binary messages.
2002/10/11	Section 2.2.7	Added note RE setting MCC/MCN octets.
2002/10/11	Section 2.2.8	Added information RE National originator.
2002/10/11	Section 2.3.4	Added note RE roaming.
2002/9/26	Section 2.2.5	Added information on invalid song titles/tempos causing RTTTL messages to be mistakenly IDed as text msgs.
2002/9/26	Section 2.2.7	Added section to explain originator rules
2002/9/26	Section 2.2.3	Added example of how to get Euro character
2002/9/26	Section 2.2.3.9	Added section to clarify timestamping
2002/5/16	Table 2-3	Removed obsolete err_detected_spam and obsolete err_rnl_failed
2002/5/16	Section 2.3.1	Added missing "and"
2002/5/16	Section 2.2.6	Added reference to Nokia Smart Messaging documents
2002/5/16	Section 2.2.5	Added missing "is"
2002/5/16	Section 2.2.3.6	Replaced "into" with "as"
2002/5/16	Section 2.2.3.3	Removed erroneous caution about MS components; replace it with a more accurate note.

Date	Section	Description of change
2002/5/16	Table 2-1	For test_mode, deleted stray "be"
2002/5/16	Section 2.2.1	"features" changed to "headers
2002/5/16	Section 2.2	Clarified that the connection URL is also the example Calling Application.
2002/5/16	Table 1-1	Added xref to Section 2.2.
2002/5/16	Section 1.1	Deleted obsolete reference to Q-Caster 2.0; added reference to Nokia Smart Messaging documents.
2002/5/8	Table 2-3	Corrected meaning of err_number_wrong_length
2002/5/8	Section 2.3	Clarified the meaning of return code="ok"
2002/5/8	Section 2.2.6	Added information RE length limitations for binary messages.
2002/5/8	Table 2-1	For originator, added information RE "callback" originators.
2002/1/17	Appendix C	Moved document change log to Appendix C.
2002/1/17	Section 2.2.3	Added information on leading \ and Ö characters and how they relate to flash messages.
2002/1/17	Section 2.2.3	Significant changes and additional subsections to clarify the use of encodings in message submission.
2002/1/17	Table 2-1	Changed regular expression to allow for dot and space characters in the originator.
2001/11/29	Section 1	References to beta status removed.
2001/11/27	Section 2.4	Removed.
2001/11/27	Listing 2-1 & Listing 2-2	Added originator attribute to <message>.
2001/11/27	Listing 2-6	Removed references to obsolete error codes.
2001/11/27	Section 2.3.3	Removed references to deprecated providercode. Corrected information regarding response to authentication failure. Removed Listing 2-5; this listing number is reserved for future use.
2001/11/27	Sections 2.3.2 & 2.3.3	Reversed order of these sections to accurately reflect the order of events.
2001/11/27	Section 2.2.6	Removed option for non-UDHI binary content. This behavior is unspecified.

Date	Section	Description of change
2001/11/27	Table 1-1	Removed references to obsolete message header and footer.
2001/11/27	Section 2.2.3	Added regular expression for text message contents.
2001/11/27	Listing 2-1 & Table 2-1	Removed reference to obsolete attribute `type`.
2001/11/27	Section 2.2.1	Added repetition of the important connection information.
2001/11/27	Section 1.1	Removed reference to deprecated providercode. Added requirement for valid IP number or range. Removed reference to ANSI X3.4-1986. Added reference to ETSI GSM 03.38. Added reference to RFC 1738.
2001/11/26	Section 2.4	Section title changed.
2001/11/26	Section 2.3.4	Removed references to obsolete `err_detected_spam`.
2001/11/26	Section 2.3.1	Corrected description of behavior of network connection.
2001/11/26	Section 2.2.5 & 2.2.6	Corrected regular expression.
2001/11/19	Section 2.1 & 2.2.1	Corrected the hostname and connection URL for the HTTP POST request.
2001/11/19	Table 2-1	Corrected information on how the number attribute is parsed and the acceptable input for this attribute.
2001/11/19	Table 2-1, Listing 2-1, Listing 2-2	Removed "brand" from required input. The brand attribute is deprecated and should be removed from the Calling Application code when convenient.
2001/11/15	Table 2-3	Added error err_invalid_content.
2001/11/15	Section 2.2.4	Clarified specifications for flash messages.
2001/11/13	Section 2.2.3	Changed requirements for encoding submissions that contain XML special characters.
2001/11/13	Appendix B	Corrected omissions in series of <note> and <duration>.
2001/11/13	Section 2.2.5	Provided more specific information about default values of RTTTL attributes.

Date	Section	Description of change
2001/10/19	Table 2-1	Character support is ETSI GSM 03.38, not US-ASCII. Binary content and RTTTL are now permitted. Long messages are now split into two messages, not truncated.
2001/10/19	Section 2.2.3	Character support is ETSI GSM 03.38, not ETSI GSM03.34. A chart of these characters and their encoding has been added.
2001/10/19	Table 2-4	Many error codes have changed. A few codes have been added.
2001/10/19	Listing 2-2	The first line of code should read: <?xml version="1.0"?>
2001/10/19	Section 2.1	Secure connections to Q-Caster 2.5 are no longer required. HTTP connections are accepted. However, 2.5 does require new usernames for all accounts.
2001/10/19	Section 2.2.4, 2.2.5, 2.2.6, and Appendix B	Added.
2001/10/19	Figure 1.1	Changed.