mGage SMS Gateway API Reference

Version: B.4.2

mGage

mGage

Document Change History

Version	Date	Changes
Document change history prior to 2013 has been removed.		
3.2	1/9/2013	Added IVR opt-in values for T-Mobile Updated MT related parameters, product_code and action are mandatory for all MTs
3.4	7/16/2013	Updated information about mTrust billing for T-Mobile
3.5	8/15/2013	Updated mTrust information for all carriers
3.6	9/6/2013	Added REVERSE functionality that allows reversing successful premium charge
3.7	9/23/2013	Updated Appendix T-Mobile Opt-in Data, removed Delivery Timestamp.
4.0	11/20/2013	Added Messaging Basics section. Added reference to new documents: Integration Pack, Technical Best Practices for Aggregation, Carrier-Feature Matrix. Updated action parameter rules. Corrected receipt-url example. Added information for receiving Multipart MO via HTTP. Removed premium references except a few controlled cases.
4.1	3/20/2014	Added seven days data retention policy note to status-url and receipt-url. Removed VMU Proxy URL service from Appendix.

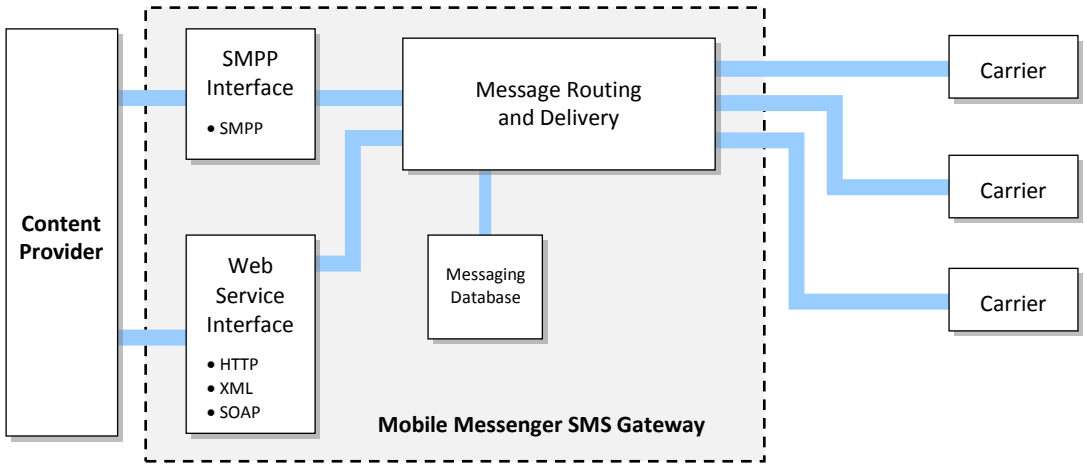
B.4.2	09/16/2015	• Updated diagrams with mGage branding and updated workflows to use correct action words • Removed references to IQ navigator help section and updated noc contact email id • Deprecated ACTION words – CANCEL, CANCELLALL, AOC, WELCOME, RENEWALREMINDER • Appendix A: Updated Carrier Support Matrix document • Removed Program types section • Updated information about window size rules • Updated information about insertion and deletion • Updated Appendix D with contact information

mGage

Contents

1. INTRODUCTION

mGage SMS Gateway is a message routing and delivery platform that gives Content Providers (CPs) access to carrier SMSCs via a carrier-grade data center. SMS Gateway automatically performs message routing to the appropriate destination carrier based on the consumer phone number. Along with support for the latest message routing standards, SMS Gateway offers 2-way standard messaging services, receipt notifications, binary content delivery, and WAP Push capabilities.



SUPPORT

Your Account Manager is the main point of contact.

- If you have any questions or issues during integration, please contact your Account Manager.
- If you have technical issues **after** you launch your program, please contact mGage NOC (24/7) :
  - Email : customer-support@mgage.com
  - Phone: +1.877.238.3637

For faster technical support response, MM clients should identify the use case clearly and provide multiple samples with the necessary details:

- Protocol used (SMPP or HTTP)
- Timestamp with time zone
- MDN
- Shortcode
- Message reference id
- Additional attributes (product_code, action, etc.)

1.1 Available SMS Gateway Interfaces

SMS Gateway supports two different messaging interfaces: SMPP and Web Service (HTTP). Since not all of the Gateway functionality is available on both interfaces, you should analyze your messaging needs carefully before committing to one interface or the other. The following table shows the feature matrix for the two interfaces.

	SMPP	Web Service
Mobile Terminated (MT) Messages	Yes	Yes
Mobile Originated (MO) Messages	Yes	Yes
WAP Push	Yes	Yes
Carrier ID in MO Messages	Yes	Yes
Delivery Receipts	Yes	Yes
Bulk Send	No	Yes
Carrier Number Lookup (this is a mandatory service for all clients)	No	Yes
Handset Lookup	No	Yes
International Delivery	Yes	Yes

1.2 Messaging Basics

This section explains high level use cases to depict how to interact with mGage system per various program types.

mGage provides a set of APIs to be used for various purposes. There is a document set that explains system capabilities and how to integrate with mGage aggregation APIs.

Necessary Documents	SMS Programs	MMS Programs
Aggregation Integration Pack The document is a startup kit that explains the documentation and integration steps.	✓	✓
SMS Gateway API Reference The document explains how to integrate with mGage SMS Messaging APIs for 2-way SMS messaging.	✓	✓
MMS Gateway API Reference The document explains how to integrate with mGage MMS Messaging APIs for 2-way MMS messaging.	☐	✓
Gateway Error Codes Error codes that come from mGage and carrier systems and their retry policies.	✓	✓
Consolidated Deactivation File This daily file includes phone numbers that are deactivated, suspended or changed phone numbers. Processing the daily deactivation files is required by the carriers.	✓	✓
Technical Best Practices for Aggregation Best practices and FAQs for integrating with mGage for messaging.	✓	✓

1.3 Content Provider Responsibilities

1.3.1 INTEGRATING WITH MGAGE

Content Providers integrating with mGage are strongly recommended to follow the Aggregation Integration Pack to ensure the integration is robust, compliant, and efficient.

1.3.2 STOP-MO

When consumer sends an SMS-MO with "STOP" (or other keywords in the STOP keyword family), Content Provider must cancel subscriptions on the shortcode and send a message to consumer with proper product code (IPC) and action=CONTENT OR HELP (either works). No more messages sent to the consumer are allowed by the carriers after this point, unless consumer interacts with the program again.

1.3.3 HELP-MO

When a consumer sends SMS-MO with “HELP”, Content Provider must send an informative SMS-MT to the consumer with proper product code (IPC) and action=HELP. For carrier rules about what to include in this message, please contact your Account Manager.

1.3.4 DEACTIVATIONS

mGage provides deactivation files to Content Providers. These deactivation files include MDNs that have been deactivated and will be reassigned to new consumers in the future. Content Providers must receive the deactivation files on a daily basis and cancel all subscriptions for the included MDNs. The deactivation files are located in IQ-Reports platform.

Some carriers do not provide deactivation files. The only way for Content Providers to keep their subscription database up to date is by cancelling subscriptions based on error codes or Delivery Receipts received from mGage. This works as follows:

1. Content Provider sends an SMS-MT to consumer. For example, daily news alerts message.
2. **mGage Rejects:** mGage rejects the SMS-MT with a severe error code.
3. **Carrier Rejects:** Alternatively, mGage accepts the SMS-MT from the Content Provider; however, in the following step the carrier rejects the SMS-MT submission. Content Provider receives a SMS Delivery Receipt for this case with a severe error code.

All error codes are included in the mGage Gateway Error Codes document. Please contact your account manager for latest Error code document.

***CRITICAL NOTE**: Content Providers with subscription programs must use delivery receipts for the carriers that do not support deactivation files. This is the only way to detect a recycled (deactivated) MDN and cancel all subscriptions. Please contact your Account Manager for the list of carriers that do not support deactivation files.*

2. SMS GATEWAY SMPP INTERFACE

The SMS Gateway SMPP interface allows messaging applications to connect with SMS Gateway via a direct bind. The SMPP interface supports 2-way SMS messages. Delivery Receipts and WAP-Push are also supported.

This chapter covers the following topics:

***NOTE**: This section of the manual assumes that you are already familiar with the SMPP protocol. For more information on SMPP, please visit Wikipedia.*

2.1 SMPP Bind Configuration

To use SMS Gateway via SMPP, you need to connect as a standard External Short Message Entity (ESME) as described in the SMPP v3.4 Specification. The preferred connection method is Transceiver, but you can also bind as a Transmitter/Receiver if necessary.

The following standard PDUs are employed for connecting, transmitting messages, and receiving messages:

Client to SMS Gateway	SMS Gateway to Client
bind_transceiver	bind_transceiver_resp
bind_transmitter	bind_transmitter_resp
bind_receiver	bind_receiver_resp
enquire_link	enquire_link
enquire_link_resp	enquire_link_resp
submit_sm	submit_sm_resp
deliver_sm_resp	deliver_sm
query_sm	query_sm_resp
unbind	unbind
unbind_resp	unbind_resp

The relevant settings are detailed below. If you need SMS Gateway to accept additional PDU types or if you require access to specific parameters, please contact the mGage Connectivity Team.

2.2 Connection Settings

Set the body parameters of the *bind_transceiver* PDU as follows:

bind_transceiver	
Parameter	**Value**
system_id	As supplied
password	As supplied
system_type	0x00
interface_version	0x34 (v3.4)
addr_ton	0x00
addr_npi	0x00
address_range	0x00

When disconnecting, always send an *unbind* PDU and wait for an unbind response before closing the socket connection. Unless you have specifically arranged multiple connections, only one SMPP bind is allowed at a time. Therefore if you don't send an *unbind* PDU, you may find that attempts to immediately rebind are denied.

Your bind should be up and ready to accept messages at all times and you have responsibility for rebinding if the connection is dropped. Note that if the connection is not available, SMS Gateway will only queue up to a certain number of MO messages (roughly 100,000). Beyond that limit, messages are discarded.

To ensure a constant connection, send an *enquire_link* PDU every 30 seconds. If you don't receive an immediate response, you should assume the connection has dropped and attempt to rebind.

2.3 Message Rate Settings

If configurable on your platform, set the message window size and MT message rate as follows:

Setting	Value	Definition
Window Size	10 msgs	The maximum number of outstanding (i.e., unacknowledged) SMPP operations that are allowed at the same time. (Window size value can be increased up to 50 based on the requirement)

MT Message Rate	10 msgs/sec	The maximum rate at which SMS Gateway is set to receive messages via SMPP. You can raise this limit by request.

If you exceed the MT message rate, SMS Gateway may reject your messages and reply with the following throttle rate error in the `command_status` header field of the *submit_sm*_resp PDU:

`0x58 – ESME_RTHROTTLED`

In this case, you should pause the message stream for one second for each error response received.

By default, the rate of MO traffic sent from SMS Gateway is unrestricted. If you need temporary MO throttling, send back the ESME_RTHROTTLED error in the *deliver_sm_resp* PDU. SMS Gateway will pause one second for each error sent. If you want to adjust the MO message rate permanently, please notify mGage.

2.4 Resubmittable Errors

In the event of an SMPP error, SMS Gateway returns a code in the `command_status` header field of the response message, as described in the SMPP 3.4 specification. The errors for which you should resubmit a message are listed in the table below, along with the suggested delay (in milliseconds), the retry limit, and the incremental delay (in milliseconds).

Event Code	Description	Delay (ms)	Limit	Inc-delay (ms)
0x0004	ESME_RINVBNDSTS	30000	10	0
0x0008	ESME_RSYSERR	10000	100	1000
0x000D	ESME_RBINDFAIL	30000	10	0
0x0011	ESME_RCANCELFAIL	30000	10	0
0x0013	ESME_RREPLACEFAIL	30000	10	0
0x0014	ESME_RMSGQFUL	10000	None	1000
0x0045	ESME_RSUBMITFAIL	10000	10	1000
0x0058	ESME_RTHROTTLED	10000	None	5000
0x0067	ESME_RQUERYFAIL	30000	10	0
0x00FE	ESME_RDELIVERYFAILURE	10000	100	1000

Any other error code means the request was invalid and you should not attempt to resubmit.

2.5 Sending MT Messages

To send standard-rated MT messages, use a regular SMPP *submit_sm* PDU with the addition of several mGage-specific parameters.

Configure the *submit_sm* as shown in the table below. Note that all other non-header parameters in the PDU must be set to NULL.

submit_sm	
Parameter	**Value**
`source_addr`	The service code you want to use for this message.
`destination_addr`	The destination number for this message. By default, SMS Gateway accepts only national format the following two formats: • 10 digits (for example, 5008885555) • 11 digits starting with 1 (for example, 15008885555) International format needs to be enabled explicitly. Unless mGage has specifically configured your SMPP bind otherwise, SMS Gateway treats these formats as North American Numbering Plan (NANP) numbers destined for a US or Canadian carrier and routes the message accordingly. **International Messages** For international messages, mGage must configure your bind to accept variable-length numbers prefixed with the country code. Refer to Appendix D: International Messages for additional information and a list of supported countries. Note that if your bind is configured for international delivery, you must use the 11-digit format shown above in order to send messages to US and Canadian carriers. If all your international messages go to a single country, mGage can optionally configure your bind so that you may omit the country code. To discuss this and other more complex international number formatting options, please contact the mGage Connectivity Team.
`esm_class`	Use only if the message is a WAP Push (see Sending WAP Push Requests). Otherwise, set to 0x00.
`registered_delivery`	Use only if you require a Delivery Receipts. Otherwise, set to 0x00.

data_coding	Use only in the following cases: ○ if the message is a WAP Push (see Sending WAP Push Requests) ○ if you need to override the default character encoding for the SMPP connection (see Appendix D: International Messages) Otherwise, set to 0x00.
sm_length	Length of the short_message data in octets.
short_message	The text of the message. The SMPP protocol supports up to 254 octets of data. However, the practical maximum message length varies by carrier. See Appendix A: Carrier Information for allowable message lengths.
product_code (0x1520)	*product_code (or referred as IPC) is an mGage-specific parameter. Required for all MT messages.* A text string that defines the program or product to which the message content belongs. Product codes are set up by mGage to meet your specific needs. The format of the product code typically indicates the Content Provider, short code, item name, pricing, and program category. For example: ACMEX_12345_RINGHIP_000_D
action (0x1581)	*mGage-specific parameter. Required for all MT messages.* A text string that indicates the type and purpose of the message. Please refer to the Actions section for the available values for this parameter.

SMS Gateway will respond with a standard *submit_sm_resp* PDU as described in the SMPP 3.4 specification.

submit_sm_resp	
Parameter	**Value**
command_status	For a successful submit, the command_status field is set to 0. For an unsuccessful submit, the field contains the relevant SMPP error. (See Resubmittable Errors above.)
message_id	For a successful submit, the message_id field contains a unique ID assigned by SMS Gateway. For an unsuccessful submit, the field is null.

***NOTE**: SMS Gateway uses the message ID returned in the submit_sm_resp PDU to reference the original message in subsequent operations such as Delivery Receipts. Especially if you plan to*

reconcile the message against the receipt, your application will need to make note of the message ID transmitted in the submit_sm_resp PDU. Please note that for the carriers that do not provide deactivation files, Content Providers with subscription programs must use Delivery Receipts in order to be notified of the deactivated (recycled) phone numbers and cancel all subscriptions for them. In general you should store the message ids for all MTs sent. Please include the ids when inquiring about MTs with us.

Standard MT Example

This sample shows the complete *submit_sm* for a standard MT message.

```
00 00 00 60 00 00 00 04 00 00 00 00 00 00 00 01 ................
00 00 00 31 32 33 34 35 00 00 00 35 30 30 38 38 ...12345...50088
38 35 35 35 35 00 00 00 00 00 00 00 00 00 00 0B 85555...........
48 65 6C 6C 6F 20 57 6F 72 6C 64 15 81 00 07 43 Hello World....C
4f 4e 54 45 4e 54 15 20 00 16 41 43 4d 45 58 5f ONTENT....ACMEX_
31 32 33 34 35 5f 4e 45 57 53 5f 30 30 30 5f 44 12345_NEWS_000_D
```

The relevant parts of the message decode as follows:

```
Command Length: 96
Command ID: 4 (submit_sm)
Source address: 12345
Destination address: 5008885555
Message length: 11
Short message: Hello World
Optional TLV 0x1520 (product_code): ACMEX_12345_NEWS_000_D
Optional TLV 0x1581 (action): CONTENT
```

Here's the *submit_sm_resp* sent back from SMS Gateway.

```
00 00 00 2d 80 00 00 04 00 00 00 00 00 00 00 08 ................
30 6a 36 74 39 75 30 31 6f 64 69 69 76 61 30 37 0j6t9u01odiiva07
6d 63 34 34 68 33 30 69 39 66 33 67 00          mc44h30i9f3g.
```

The relevant parts of the message decode as follows:

```
Command length: 45
Command ID: 80000004 (submit_sm_resp)
Command status: 0 (no error)
Message ID: 0j6t9u01odiiva07mc44h30i9f3g
```

2.6 Message Receipts

The SMS Gateway SMPP interface allows you to request receipts for standard rate MT messages. This feature is implemented using the standard SMPP delivery receipt functionality. Please refer to the SMPP Specification v3.4 for a complete description of delivery receipt behavior on 3.4 connections.

***CRITICAL NOTE**: Content Providers with subscription programs must use delivery receipts for the carriers that do not support deactivation files. This is the only way to detect a recycled (deactivated) MDN and cancel all subscriptions. Please contact your Account Manager for the list of carriers that do not support deactivation files.*

2.6.1 AVAILABLE RECEIPT TYPES

SMS Gateway supports three types of message receipts: Delivery, Failure, and Enroute. The availability of these receipts is dependent on the policy of the individual carriers.

Enroute Receipts

Enroute Receipts are generated by SMS Gateway to indicate that the destination carrier SME gateway accepted the message.

Handset Delivery Receipts

Handset Delivery Receipts are generated when the carrier delivers the message to the subscriber. In most cases if the message is immediately received by the handset, or if the message is rejected by the carrier, you'll receive the receipt within a few seconds. However, depending on the carrier's message validity period, the Delivery Receipt may not be sent for up to 72 hours.

Failure Receipts

Failure Receipts are generated when the carrier cannot deliver the message to the subscriber, or when SMS Gateway cannot submit the message to the carrier SME gateway. Carrier failures may take up to 72 hours to come back, depending on the message validity period.

2.6.2 REQUESTING A MESSAGE RECEIPT

To request message receipts from SMS Gateway, set the optional `registered_delivery` parameter in the *submit_sm* PDU as indicated in the table below.

submit_sm		
Parameter	**Value**	**Receipt Type**
`registered_delivery`	0x00	No receipt
	0x01	Delivery Receipt (or Failure Receipt) You'll get a receipt when the message is delivered to the handset (or marked as undeliverable) by the carrier.
	0x02	Failure Receipt ONLY You'll get a receipt if SMS Gateway rejects the message or the message is marked as undeliverable by the carrier.

0x04	Enroute Receipt ONLY You'll get a receipt as soon as the message is accepted or rejected by carrier.
0x05	Enroute Receipt AND Delivery Receipt (or Failure Receipt) You'll get a receipt as soon as the message is accepted or rejected by carrier. If the message is accepted, you'll get a second receipt when the message is delivered (or marked as undeliverable) to the handset.
0x06	Enroute Receipt AND Failure Receipt You'll get a receipt as soon as the message is accepted or rejected by the carrier. If the message is later marked as undeliverable by the carrier, you'll also receive a failure receipt.

***NOTE**: In accordance with Verizon Wireless policy, SMS Gateway must "upgrade" the `registered_delivery` parameter to 0x01 on messages sent to Verizon Wireless subscribers. Even if you set `registered_delivery` to 0x00 or 0x04, you'll still get a Delivery Receipt or a Failure Receipt.*

Example

This sample shows the complete *submit_sm* for a standard MT message with Delivery Receipts requested.

```
00 00 00 60 00 00 00 04 00 00 00 00 00 00 00 01 ................
00 00 00 31 32 33 34 35 00 00 00 35 30 30 38 38 ...12345...50088
38 35 35 35 35 00 00 00 00 00 00 01 00 00 00 0B 85555...........
48 65 6C 6C 6F 20 57 6F 72 6C 64 15 81 00 07 43 Hello World....C        4f 4e
54 45 4e 54 15 20 00 16 41 43 4d 45 58 5f ONTENT....ACMEX_
31 32 33 34 35 5f 4e 45 57 53 5f 30 30 30 5f 44 12345_NEWS_000_D
```

The relevant parts of the message decode as follows:

```
Command Length: 96
Command ID: 4 (submit_sm)
Source address: 12345
Destination address: 5008885555
Registered delivery: 1
Message length: 11
Short message: Hello World
Optional TLV 0x1520 (product_code): ACMEX_12345_NEWS_000_D
Optional TLV 0x1581 (action): CONTENT
```

2.6.3 RECEIVING A MESSAGE RECEIPT

Message receipts are delivered from SMS Gateway via a standard *deliver_sm* PDU with the `esm_class` field set to indicate that the `short_message` field contains a standard SMSC delivery receipt as defined in Appendix B of the SMPP 3.4 specification.

Relevant status information is returned via the `esm_class`, `message_state`, and `receipted_message_id` parameters in the *deliver_sm* PDU.

deliver_sm	
Parameter	**Value**
`esm_class`	Indicates the source of the message receipt: 0x04 The receipt originated from a carrier (Delivery Receipts and Failure Receipts). 0x08 The receipt originated from SMS Gateway (Enroute Receipts and Failure Receipts).
`message_state` (0x0427)	Indicates the overall status of the message. There are eight possible message states defined in the SMPP specification: 0x01 (Enroute) Message is in transit. 0x02 (Delivered) Message has been successfully delivered. 0x03 (Expired) Message validity period has expired. 0x04 (Deleted) Message has been deleted. 0x05 (Undeliverable) Message is undeliverable. 0x06 (Accepted) Message is in an accepted state (i.e., has been manually read on behalf of the subscriber by customer service). 0x07 (Unknown) Message is in an invalid state. 0x08 (Rejected) Message has been rejected. Please note that this TLV tag is only included by a few carriers (e.g. US Cellular). For carriers who do not include this TLV tag you will have to parse the message text of the DRs to evaluate the delivery status.
`receipted_message_id` (0x001E)	Matches the ID indicated in the *submit_sm_resp* that SMS Gateway sent in response to the original message. Used to reconcile the receipt with the message to which it applies.

The Delivery Receipt contained in the `short_message` parameter includes the eight standard fields defined in the SMPP specification, plus an additional mGage field (`idd`). For example:

```
id:0fkdjrt00g003v0h7k4351h4u2ds sub:001 dlvrd:001 submit date:0810201840
done date:0810201841 stat:ENROUTE err:000 text:hello wireless world
idd:VZW^11d3f64ac9d61
```

short_message			
Field	**Length**	**Field Type**	**Value**
`id`	28	Standard	*If `esm_class`=0x08 (SMS Gateway-generated receipt):* Set to the same value as the `receipted_message_id` (0x001E) parameter (see above). *If `esm_class`=0x04 (carrier-generated receipt):* Contains the carrier's internal ID for the receipt message.
`sub`	3	Standard	Always set to 001 for message receipts.
`dlvrd`	3	Standard	Always set to 001 for message receipts.
`submit date`	10	Standard	The message submit date in YYMMDDmmhh format.
`done date`	10	Standard	The message completion date in YYMMDDmmhh format.

`stat`	7	Standard	A text version of the `message_state` (0x0427) parameter (see above). Possible values are as follows:
			`ENROUTE`
			`DELIVRD`
			`EXPIRED`
			`DELETED`
			`UNDELIV`
			`ACCEPTD`
			`UNKNOWN`
			`REJECTD`

err	3	Standard	Set to 000 for Enroute Receipts and Delivery Receipts. For Failure Receipts, contains a carrier-specific error code or an SMS Gateway error code indicating the cause of the failure. (See Appendix B: SMS Gateway Error Codes)
text	12	Standard	The first 12 characters of the original message.
idd	3	mGage	The idd field is added to SMS Gateway-generated receipts (esm_class=0x08*)* to indicate the ID of the original message received from the carrier (where available). For Enroute Receipts, idd is set to the carrier's internal ID for the receipt message. For Failure Receipts, idd is set to "null." The field is omitted on carrier-generated Delivery or Failure receipts (esm_class=0x04).

Enroute Receipt Example

This sample shows the complete *deliver_sm* for an Enroute Receipt.

```
00 00 00 ec 00 00 00 05 00 00 00 00 00 00 00 06 ...y............
00 02 01 35 30 30 38 38 38 35 35 35 35 00 00 09 ...5008885555...
31 32 33 34 35 00 08 00 00 00 00 00 00 00 00 96 12345..........?
69 64 3a 31 70 39 37 66 71 6a 30 37 37 65 6a 64 id:1p97fqj077ejd
61 32 6c 73 65 70 61 39 35 6e 38 63 6e 75 75 20 a2lsepa95n8cnuu
73 75 62 3a 30 30 31 20 64 6c 76 72 64 3a 30 30 sub:001 dlvrd:00
31 20 73 75 62 6d 69 74 20 64 61 74 65 3a 30 38 1 submit date:08
31 30 32 39 32 31 30 37 20 64 6f 6e 65 20 64 61 10292107 done da
74 65 3a 30 38 31 30 32 39 32 31 30 37 20 73 74 te:0810292107 st
61 74 3a 45 4e 52 4f 55 54 45 20 65 72 72 3a 30 at:ENROUTE err:0
30 30 20 74 65 78 74 3a 4d 54 6d 65 73 73 61 67 00 text:MTmessag
65 20 69 64 64 3a 56 5a 57 5e 31 31 64 34 61 36 e idd:VZW^11d4a6
65 38 37 32 64 31 00 1e 00 1d 31 70 39 37 66 71 e872d1....1p97fq
6a 30 37 37 65 6a 64 61 32 6c 73 65 70 61 39 35 j077ejda2lsepa95
6e 38 63 6e 75 75 00 04 27 00 01 01             n8cnuu..'...
```

The relevant parts of the message decode as follows:

```
Command Length: 236
Command ID: 5 (deliver_sm)
Source address: 5008885555
Destination address: 12345
ESM class: 8
```

```
Optional TLV 0x001E (receipted_message_id): 1p97fqj077ejda2lsepa95n8cnuu
Optional TLV 0x0427 (message_state): 1
Message: id:1p97fqj077ejda2lsepa95n8cnuu sub:001 dlvrd:001
         submit date:0810292107 done date:0810292107 stat:ENROUTE
         err:000 text:MTmessage idd:VZW^11d4a6e872d1
```

Delivery Receipt Example

This sample shows the complete *deliver_sm* for a Delivery Receipt on a standard-rated message.

```
00 00 00 cb 00 00 00 05 00 00 00 00 00 00 00 01 ...A............
00 00 00 35 30 30 38 38 38 35 35 35 35 00 00 00 ...5008885555...
31 32 33 34 35 00 04 00 00 00 00 00 00 00 00 7e 12345..........~
69 64 3a 56 5a 57 5e 31 31 64 34 65 35 33 31 63 id:VZW^11d4e531c
38 65 30 20 73 75 62 3a 30 30 31 20 64 6c 76 72 8e0 sub:001 dlvr
64 3a 30 30 31 20 73 75 62 6d 69 74 20 64 61 74 d:001 submit dat
65 3a 30 38 31 30 33 30 31 35 31 35 20 64 6f 6e e:0810301515 don
65 20 64 61 74 65 3a 30 38 31 30 33 30 31 35 31 e date:081030151
35 20 73 74 61 74 3a 44 45 4c 49 56 52 44 20 65 5 stat:DELIVRD e
72 72 3a 30 30 30 20 74 65 78 74 3a 4d 54 6d 65 rr:000 text:MTme
73 73 61 67 65 00 1e 00 1d 31 62 64 68 64 63 65 ssage....1bdhdce
30 36 68 71 69 32 38 32 34 71 31 72 72 6d 37 6d 06hqi2824q1rrm7m
74 72 39 65 64 00 04 27 00 01 02                 tr9ed..'...
```

The relevant parts of the message decode as follows:

```
Command Length: 203
Command ID: 5 (deliver_sm)
Source address: 5008885555
Destination address: 12345
ESM class: 4
Optional TLV 0x001E (receipted_message_id): 1bdhdce06hqi2824q1rrm7mtr9ed
Optional TLV 0x0427 (message_state): 2
Message: id:VZW^11d4e531c8e0 sub:001 dlvrd:001
         submit date:0810301515 done date:0810301515 stat:DELIVRD
         err:000 text:MTmessage
```

Failure Receipt Example

This sample shows the complete *deliver_sm* for a Failure Receipt generated by SMS Gateway.

```
00 00 00 e0 00 00 00 05 00 00 00 00 00 00 00 03 ...m............
00 00 00 35 30 30 38 38 38 35 35 35 35 00 00 00 ...5008885555...
31 32 33 34 35 00 08 00 00 00 00 00 00 00 00 8a 12345..........?
69 64 3a 30 6e 33 39 64 34 76 32 6e 33 61 6a 6b id:0n39d4v2n3ajk
71 32 6b 76 62 6c 34 64 35 65 76 6b 36 30 69 20 q2kvbl4d5evk60i
73 75 62 3a 30 30 31 20 64 6c 76 72 64 3a 30 30 sub:001 dlvrd:00
31 20 73 75 62 6d 69 74 20 64 61 74 65 3a 30 38 1 submit date:08
31 30 33 30 31 35 31 36 20 64 6f 6e 65 20 64 61 10301516 done da
74 65 3a 30 38 31 30 33 30 31 35 31 36 20 73 74 te:0810301516 st
61 74 3a 55 4e 44 45 4c 49 56 20 65 72 72 3a 30 at:UNDELIV err:0
30 31 20 74 65 78 74 3a 4d 54 6d 65 73 73 61 67 01 text:MTmessag
65 20 69 64 64 3a 6e 75 6c 6c 00 1e 00 1d 30 6e e idd:null....0n
33 39 64 34 76 32 6e 33 61 6a 6b 71 32 6b 76 62 39d4v2n3ajkq2kvb
6c 34 64 35 65 76 6b 36 30 69 00 04 27 00 01 05 l4d5evk60i..'...
```

The relevant parts of the message decode as follows:

```
Command Length: 224
Command ID: 5 (deliver_sm)
Source address: 5008885555
Destination address: 12345
ESM class: 8
Optional TLV 0x001E (receipted_message_id): 0n39d4v2n3ajkq2kvbl4d5evk60i
Optional TLV 0x0427 (message_state):
Message: id:0n39d4v2n3ajkq2kvbl4d5evk60i sub:001 dlvrd:001
         submit date:0810301516 done date:0810301516 stat:UNDELIV
         err:001 text:MTmessage idd:null
```

2.6.4 SENDING WAP PUSH REQUESTS

If your connection is provisioned to support it, you can send WAP Push requests via SMPP across participating carriers. SMS Gateway will recognize the request and execute the appropriate carrier-specific WAP Push on your behalf.

Some carriers also have additional restrictions on WAP pushes. See Appendix C: Carrier-Specific Notes for more information.

***NOTE**: SMS Gateway supports WAP Push via SMPP primarily to accommodate SMPP-centric organizations. If your messaging applications do not need to be exclusively SMPP-based, the Web Service interface provides a simpler WAP Push mechanism. Please see Chapter 3: SMS Gateway Web Service Interface for details.*

This section provides an overview of how to send a WAP Push via SMS Gateway using SMPP. For more detailed background information, please consult the following WAP Specification documents:

- WAP Service Indication Specification (WAP-167-ServiceInd-20010731-a)
- Wireless Session Protocol Specification (WAP-230-WSP-20010705-a)
- Wireless Datagram Protocol Specification (WAP-259-WDP-20010614-a)

These documents are available in PDF format from the Open Mobile Alliance website:

http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

CONSTRUCTING THE WAP PUSH

An SMS Gateway WAP Push is achieved by placing a binary encoded Service Indication (SI) into the `short_message` parameter of a *submit_sm* PDU. The message is comprised of four main pieces.

submit_sm Parameters
User Data Header
WTP Header
Service Indication

***NOTE**: The current version of SMS Gateway requires WAP Push requests to be encoded in a single SMPP message. This limits the combined size of the title and the WAP URL.*

SUBMIT_SM PARAMETERS

Set the *submit_sm* PDU parameters normally as described in the Sending MT Messages section, with the following additions:

- Set the `esm_class` parameter to 0x40. This turns on the User Data Header Indicator (UDHI) flag, indicating that the beginning of the `short_message` data is a header. (See section 5.2.12 of the SMPP v3.4 Specification.)
- Set the `data_coding` parameter to 0xF4. This changes the SMPP data coding scheme of the `short_message` contents to GSM message class control, 8-bit binary. (See section 5.2.19 of the SMPP v3.4 Specification.)
- Set the mGage-specific `action` TLV (0x1581) to `WAPPUSH`.

For example:

```
00 00 00 60 00 00 00 04 00 00 00 00 00 00 00 01
00 00 00 31 32 33 34 35 00 00 00 35 30 30 38 38
38 35 35 35 35 00 40 00 00 00 00 00 00 f4 00
```

USER DATA HEADER

The `short_message` parameter must begin with a User Data Header (UDH) string that defines the destination port as 2948 and the source port as 9200:

```
06 05 04 0B 84 23 F0
```

The seven octets translate as:

06 = UDH length
05 = Identifier for application port addresses
04 = Header length
0B 84 = Destination port 2948
23 F0 = Source port 9200

See WAP-259-WDP-20010614-a for additional information about how User Data Headers are used in WAP.

WTP Header

The UDH is followed by a WTP header string. For example:

```
02 06 01 ae
```

The four octets translate as:

02 = Transaction ID (Push ID)
06 = PDU type (Push PDU)
01 = Header length
AE = Well-known header value token for “Content-type: application/vnd.wap.sic”

See WAP-230-WSP-20010705-a, section 8.2 “Protocol Data Unit Structure” for more information.

Service Indication

The WTP header is followed by a Service Indication containing the URL and message text, tokenized using WBXML encoding. For example, this Service Indication XML yields the WBXML string shown below.

```
<?xml version="1.0"?>
<!DOCTYPE si PUBLIC "-//WAPFORUM//DTD SI 1.0//EN"
"http://www.wapforum.org/DTD/si.dtd">
<si>
<indication href="http://wap.yahoo.com">m-Qube Msg</indication>
</si>
```

```
03 05 6a 00 45 C6 0C 03 77 61 70 2E 79 61 68 6F 6F 2E 63 6F 6D 00 01 03
6D 2D 51 75 62 65 20 4D 73 67 00 01 01
```

See WAP-167-ServiceInd-20010731-a for more information about Service Indications and WBXL encoding.

WAP Push Example

This sample shows the *submit_sm* parameters, User Data Header, WTP header, and Service Indication binary combined to form the complete SMS.

```
00 00 00 60 00 00 00 04 00 00 00 00 00 00 00 01 ................
00 00 00 31 32 33 34 35 00 00 00 35 30 30 38 38 ...12345...50088
38 35 35 35 35 00 40 00 00 00 00 00 00 f4 00 30 85555.@........0
06 05 04 0b 84 23 f0 02 06 01 ae 03 05 6a 00 45 .....#.........j.E
c6 0c 03 77 61 70 2e 79 61 68 6f 6f 2e 63 6f 6d ...wap.yahoo.com
00 01 03 6d 2d 51 75 62 65 20 4d 73 67 00 01 01 ...m-Qube Msg...
```

2.7 Receiving MO Messages

For the SMPP interface, SMS Gateway delivers MO messages via a standard *deliver_sm* PDU configured as follows:

deliver_sm	
Parameter	**Value**
source_addr	The mobile number from which the message originated.
destination_addr	The service code used for the message.
sm_length	Length of the short_message data in octets.
short_message	The text of the message.

To enable you to identify the carrier, the *deliver_sm* also includes the mGage-specific optional carrier_id parameter:

deliver_sm	
Parameter	**Value**
carrier_id (0x1402)	String representing the numerical value of the carrier ID. See Appendix A: Carrier Information for the full list of carrier IDs.

Example

This sample shows an MO delivered from carrier 2 (T-Mobile).

```
00 00 00 4B 00 00 00 05 00 00 00 00 00 00 00 01 ................
00 00 00 35 30 30 38 38 38 35 35 35 35 00 00 00 ...5008885555...
31 32 33 34 35 00 00 00 00 00 00 00 00 00 00 0B 12345...........
48 65 6C 6C 6F 20 57 6F 72 6C 64 14 02 00 01 32 Hello World....2
14 04 00 07 54 4d 6f 62 69 6c 65                ....Tmobile
```

The relevant parts of the message decode as follows:

```
Command length:75
Command ID: 5 (deliver_sm)
Source address: 5008885555
Destination address: 12345
Message: Hello World
Optional TLV 0x1402 (carrier_id): 2
```

2.7.1 ACKNOWLEDGING MO MESSAGES

You must reply with a *deliver_sm_resp* PDU for all MO messages sent to you. If a message is unacceptable to your system, please reply with an appropriate SMPP error code. If you do not reply with a *deliver_sm_resp*, SMS Gateway will continue to attempt delivery only until the queue reaches a certain size. (See Connection Settings.)

3. SMS GATEWAY WEB SERVICE INTERFACE

The mGage SMS Gateway Web Service interface supports both individual and bulk send transmission of MT messages. Delivery receipts and WAP-Push are also available.

SMS Gateway also hosts an API, called checkMobileNumber, that returns carrier information, status, and device make and model (for supporting carriers) for a phone number.

This chapter covers the following topics:

3.1 SMS Gateway Web Service Overview

The interface is comprised of five separate Web services, each of which is described in detail in this chapter.

Web Service	Description	Call Methods
`sendJob()`	Receives and processes bulk send job requests. See Sending Bulk MT Messages.	HTTP POST
`checkJobStatus()`	Checks to see if a bulk send job is already in the queue. See Avoiding Duplicate Job Submissions.	HTTP GET
`getJobConfig()`	Shows the current field size limits for a bulk send job request. See Checking Field Size Limits.	HTTP GET
`sendSingle()`	Sends individual MT messages. See Sending Individual MT Messages.	HTTP POST/GET SOAP
`checkMobileNumber()`	Returns the carrier ID associated with a specified mobile number and optionally provides details about the handset. See Checking a Mobile Number.	HTTP POST/GET SOAP

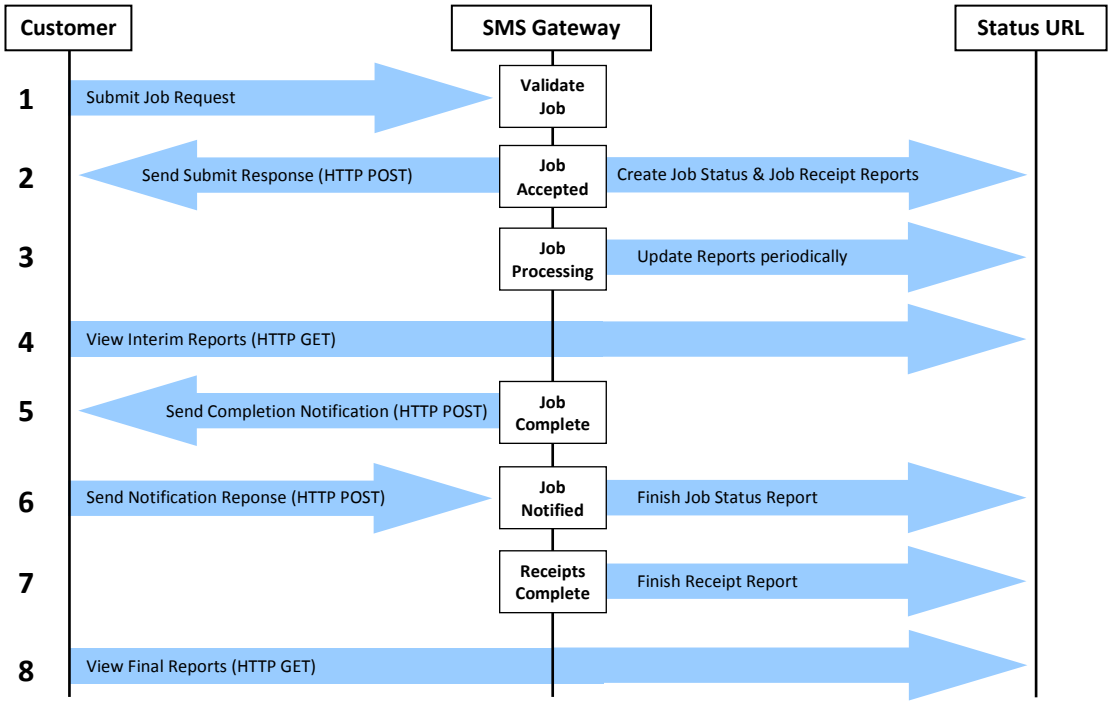
3.2 Sending Bulk MT Messages

In order to efficiently handle high-volume MT message traffic, the SMS Gateway Web Service interface includes a bulk send mechanism that organizes individual messages into units of work called jobs. Each bulk send job is defined by a Job Request, which consists of a text message, optional parameters, and a list of recipients. You submit the Job Request as an XML document via an HTTP POST.

To allow you to monitor the progress of a bulk send job, SMS Gateway makes available two status reports - one showing message delivery and one showing message receipts. You can view these reports using a standard Web browser, or download and parse them with a program. When the job is complete, SMS Gateway notifies you by posting to a provided URL.

Bulk Send Job Lifecycle

This diagram illustrates the order of events in a typical bulk send job.



Step 1
You submit a Job Request to SMS Gateway via an HTTP post.

Step 2
SMS Gateway receives the request, validates it, and returns a submit response containing a unique URL where you can view the status reports. (See Job Submit Response for details.) Two XML status reports are available for each job: the Job Status Report tracks the success (or failure) of each message and the Job Receipt Report shows any delivery receipts received in response to the messages.

Step 3
SMS Gateway begins processing the messages, updating the two reports to reflect the job's progress as it goes.

Step 4
During job processing, you can optionally monitor the reports - either using an automated program or by loading the URLs into a browser and inspecting the documents manually.

Step 5
When it determines that the job has finished, SMS Gateway marks the Job Status Report as complete and sends a notification via HTTP POST to a receiving program at a predefined URL of your choice. (See Job Notification for more information.)

Step 6
The program receiving the notification acknowledges with an HTTP POST.

Step 7
To give all carrier receipts time to return, SMS Gateway continues to update the Job Receipt Report for up to 72 hours after the completion of the job.

Step 8
The finished Job Status Report and Job Receipt Report are available for viewing.

3.2.1 BUILDING THE JOB REQUEST

The first step in the bulk send process is to create the Job Request that defines the SMS content, sets any optional job parameters, and defines the list of recipients. The Job Request is an XML document containing the following parameters:

Name	Definition
`<job-request-id>`	A Job Request ID that SMS Gateway can use to identify this job. The ID can be up to 64 characters long, should use alphanumeric characters only, and must be globally unique. To ensure that the string is unique, mGage suggests that you prepend the first few characters of your organization ID and embed a timestamp.
`<service-code>`	The short code you want to use for these messages.
`<recipient-count>`	The number of message recipients included in the job. The current maximum is 1000.
`<notification-url>` (optional)	The URL to which SMS Gateway should send the completion notification. For example: `http://mysite.com/jobComplete.jsp` See Job Notification for more information.
`<message>`	The default message used for this job. You can optionally place a `<message>` tag inside individual recipient tags to override the default.

<sms>	The actual message content. For non-text messages, the <sms> parameter accepts an optional encoding attribute to indicate the content type. The following values are supported: <sms encoding="binary_base64"> Base64 encoded binary message <sms encoding="binary_base64_UDH"> Base64 encoded binary message with header information
<receipt-options> (optional)	**CRITICAL NOTE: Content Providers with subscription programs must use delivery receipts for the carriers that do not support deactivation files. This is the only way to detect a recycled (deactivated) MDN and cancel all subscriptions. Please contact your Account Manager for the list of carriers that do not support deactivation files.** The type of receipt that you want to appear on the Job Receipt Report. The following options are available: SEND_FAIL This notification is only sent when carrier side rejects the submission of the message. This is the default receipt type for the Job Receipt Report and will appear even if you omit the <receipt-options> parameter. DELIVERED This notification is supported for designated carriers only. This notification is sent when a handset Delivery Receipt is sent by the consumer's device acknowledging it has received the message. DELIVER_NOTIFY This notification is a combination of the two receipt options above with an additional case. It is sent when: • The carrier accepts or rejects the submission of the message. • And if the submission to the carrier is successful, a following handset delivery notification will be sent for supporting carriers only. See Job Receipt Report for more information.

<product-code>	*mGage-specific parameter. Required for all MT messages.* A text string that defines the program or product to which the message content belongs. Product codes are set up by mGage to meet your specific needs. The format of the product code typically indicates the content provider, short code, item name, pricing, and program category. For example: ACMEX_12345_RINGHIP_000_D You can optionally place a <product-code> tag inside individual <r> tags to override the default.
<action>	*mGage-specific parameter. Required for all MT messages.* A text string that indicates the type and purpose of the message. Please refer to the Actions section for the available values for this parameter. You can optionally place an <action> tag inside individual <r> tags to override the default.
<recipients>	The complete set of recipients.
<r>	Individual recipients are bounded by an <r> element. You can optionally place <product-code>, <action>, and <message> tags inside individual <r> elements to override the defaults.

<destination>	The recipient's mobile number, formatted in one of the following ways: • a 10-digit number (for example, 5008885555) SMS Gateway will treat this as a North American Numbering Plan (NANP) number destined for a US or Canadian carrier. • an 11-digit number that starts with 1 (for example, 15008885555) SMS Gateway will treat this as a NANP number destined for a US or Canadian carrier. • a variable-length number prefixed with + (for example, +447700888555) SMS Gateway will treat this as an international number that includes a country code prefix. See Appendix D: International Messages for additional information and a list of supported countries. Note that you can optionally include a tel: UMDA prefix for compatibility with earlier versions of SMS Gateway. For example: tel:5008885555 The maximum number of destinations allowed in a single job is controlled by the maxRecipientsPerXmlJob parameter in the SMS Gateway job configuration file. (See Checking Field Size Limits.) Currently the maximum is 1000 destinations per job.

Example Job Request XML

Note that in this example the default SMS text and product code are overridden for the last recipient.

```
<job-request>
 <job-request-id>abc234354659234</job-request-id>
 <service-code>12345</service-code>
 <receipt-options>DELIVERED</receipt-options>
 <notification-url>http://mysite.com/jobComplete.jsp</notification-url>
 <recipient-count>3</recipient-count>
 <message>
  <sms>Two guys go into a bar...</sms>
 </message>
 <product-code>ACMEX_12345_JOKE_000_D</product-code>
 <action>CONTENT</action>
 <recipients>
  <r>
   <destination>tel:6175551000</destination>
  </r>
  <r>
   <destination>tel:6175551001</destination>
  </r>
  <r>
```

```
    <destination>tel:6175551002</destination>
    <!-- override the default message -->
    <message>
     <sms>A giraffe goes into a bar...</sms>
    </message>
    <!-- override the default product-code -->
    <product-code>ACMEX_12345_JOKE_000_D</product-code>
   </r>
  </recipients>
</job-request>
```

3.2.2 SUBMITTING THE JOB REQUEST

Once you've created the Job Request XML document, submit it via HTTP POST to the `sendJob()` Web service located at the following URL:

`https://sendsms.mobilemessenger.com/wsgw/sendJob`

The POST must send a username and password via basic authentication and should consist of a single content parameter:

`JobXML=`*xml*

where *xml* is the Job Request XML document.

3.2.3 JOB SUBMIT RESPONSE

SMS Gateway replies with a parser-friendly submit response for each Job Request it receives. The response contains the following information:

Name	Description
`job-request-id`	The unique Job Request ID you submitted in the request XML document.
`mqube-id`	A mGage job ID added by SMS Gateway.
`accepted-date`	Date and time the job was accepted.
`status-code`	The status code for the job submission. See Job Submission Status Codes for a list of possible return codes.
`status-details`	A description of the status code.
`max-retries`	The suggested number of retries that should be attempted if the status code is an error.
`retry-delay-secs`	The suggested retry delay if the status code is an error.
`num-umdas-accepted`	The number of message recipients detected in the job.

status-url	The URL where the Job Status Report is available. The URL is a combination of your mGage organization ID, a time stamp, and the two job IDs: `https://status.mobilemessenger.com/status/gws/<orgID>/SMS/<timestamp>/<job-request-id>-<mqube-id>.xml`
receipt-url	The URL where the Job Receipt Report is available. The URL is a combination of your mGage organization ID, a time stamp, and the two job IDs: `https://status.mobilemessenger.com/status/gws/<orgID>/SMS/<timestamp>/<job-request-id>-<mqube-id>-receipts.xml`

***NOTE**: The data retention policy for status-url and receipt-url is seven days. Querying messages that are older than seven days will not return status or receipt.*

Example Job Submit Response

```
job-request-id=abc234354659234
mqube-id=08urnjq00g003v0bk246419epvlk
accepted-date=05/22/2007 6:15:35 PM UTC
status-code=0
status-details=Job Accepted
max-retries=0
retry-delay-secs=0
num-destinations-accepted=3
status-url=https://status.mobilemessenger.com/status/gws/7fdhts45y4349
08ksl78m21d8641/SMS/2007052218/08urnjq00g003v0bk246419epvlk-abc2343546592
34.xml
receipt-url=https://status.mobilemessenger.com/status/gws/7fdhts45y434
908ksl78m21d8641/SMS/2007052218/abc234354659-23408urnjq00g003v0bk246419epvlk-
receipts.xml
```

3.2.4 JOB SUBMISSION STATUS CODES

The following table lists the possible status codes returned by SMS Gateway in response to a Job Request.

Code	Error Condition	Retry Logic
0	Job Accepted. The Job Request XML was successfully submitted for processing.	Do not retry
2	Invalid parameter value. The `status-details` field in the Job Submit Response contains details about the error.	Stop message processing

3	XML parsing error. The `status-details` field in the Job Submit Response contains details about the error.	Stop message processing
4	Duplicate request ID. The Job Request ID must be globally unique.	Stop message processing
5	Job processing error	The `max-retries` field in the Job Submit Response suggests the appropriate retry policy.
6	Node not accepting	The `max-retries` field in the Job Submit Response suggests the appropriate retry policy
	General HTTP errors (connection timeouts, etc.)	Retry as appropriate after first making sure the job wasn't submitted prior to the error occurring. (See Avoiding Duplicate Job Submissions.)

3.2.5 AVOIDING DUPLICATE JOB SUBMISSIONS

If a connection timeout or other HTTP error occurs before you receive the Job Submit Response, you should check to see if the job was successfully placed into the queue before you resubmit the Job Request. If you resubmit a job unintentionally you risk duplicate processing of messages.

To see if a job is already in the queue, send an HTTP GET containing the Job Request ID to the `checkJobStatus()` Web service method:

```
GET https://ws.mobilemessenger.com/wsgw/checkJobStatus?JobRequestID
=abc234354659234
```

If the job already exists, you'll receive a copy of the original Job Submit Response (see Job Submit Response). If the job doesn't exist, the call returns "INVALID" - which means it's safe to go ahead and submit.

3.2.6 CHECKING FIELD SIZE LIMITS

Size limits for the various data fields in a Job Request are maintained remotely by mGage. If any of your Job Request parameters exceed these size limits, the job will fail with status code 2 and the `status-details` field will indicate which value was at fault.

You can retrieve the current list of maximum field lengths by sending a simple HTTP GET to the `getJobConfig()` Web service method:

```
GET https://ws.mobilemessenger.com/wsgw/getJobConfig
```

Basic authentication is required to access the page, which returns data in key/value pairs as follows:


```
maxRecipientsPerXmlJob=1000
serviceCodeSizeMax=18
```

```
messageSizeMax=496
urlSizeMax=1024
proxyHostSizeMax=255
proxyPortSizeMax=5
jobRequestIdSizeMax=64
```

3.3 Tracking the Progress of a Job

Once you've submitted a Job Request to SMS Gateway you can monitor message delivery progress via two online reports - the Job Status Report and the Job Receipt Report. As soon as the job is complete, SMS Gateway also sends you a Job Notification to which you must respond.

3.3.1 JOB STATUS REPORT

The Job Status Report tracks the success (or failure) of each individual message as they are being sent from the Content Provider to mGage. Job Receipt Report is the report that displays the status of the MT submitted to the carrier.

The report is generated as soon as the job is accepted and is updated continuously during message processing. The unique URL for the report is defined in the Job Submit Response you received when you submitted the Job Request.

The following XML elements are returned in the report document:

Element	Description
`<job-request-id>`	The unique Job Request ID you submitted in the request XML document.
`<mqube-id>`	The mGage job ID added by SMS Gateway.
`<recipient-count>`	The number of message recipients in the job.
`<nodeid>`	An internal mGage processing parameter.

`<status>`	The first `<status>` element in the report indicates the status of the job as a whole. The return values are as follows:
	`RECEIVED`
	SMS Gateway received and accepted the job and message processing is underway.
	`COMPLETE`
	All messages have either been sent, failed, or retried the maximum number of times.
	`CANCELLED`
	The job was stopped before completion and will not be retried.

`<recipients>`	The complete set of recipients.

<recipient>	Each individual recipient is bounded by a <recipient> element.
<destination>	The recipient's mobile number.
<msgid>	The message ID assigned by SMS Gateway.
<status>	The status of the individual message. Possible values are: UNSENT The message is still in the queue. SENT The message was successfully sent. RETRIABLE The message could not be sent but can be retried. This usually represents a service availability error. FAIL The message could not be sent.
<error-message>	If the message status is FAIL, an additional element appears in the report to list the error code (when available). See Appendix B: SMS Gateway Error Codes.

Example Job Status Report

```
<job-report>
 <job-request-id>abc234354659234/job-request-id>
 <mqube-id>08urnjq00g003v0bk246419epvlk</mqube-id>
 <recipient-count>3</recipient-count>
 <nodeid>7</nodeid>
 <status>COMPLETE</status>
 <recipients>
  <recipient>
   <destination>6175551000</destination>
   <msgid>0fjkjrt00g003v0b7k4351h4u2eg</msgid>
   <status>SENT</status>
  </recipient>
  <recipient>
   <destination>6175551001</destination>
   <msgid>1fsluoh00g003v0b7k4351h4u2dg</msgid>
   <status>FAIL</status>
   <error-message>-20700</error-message>
  </recipient>
  <recipient>
   <destination>6175551002</destination>
   <msgid>16q21pe00g003v0b7k4351h4u2a0</msgid>
   <status>SENT</status>
  </recipient>
 </recipients>
</job-report>
```

3.3.2 JOB RECEIPT REPORT

The Job Receipt Report shows delivery receipts received for the messages sent as part of the job. The report is generated as soon as the Job Request is accepted and is updated continuously during message processing. The unique URL for the report is defined in the Job Submit Response you received when you submitted the Job Request.

The following XML elements are returned in the report document:

Element	Description
<job-request-id>	The unique Job Request ID you submitted in the request XML document.
<mqube-id>	The mGage job ID added by the Gateway.
<receipts>	The complete set of receipts for this job.
<receipt>	An individual message receipt.
<id>	The message ID of the receipt.
<msgid>	The ID of the original message the receipt is in response to.
<delivered>	If the message was delivered to the handset, the receipt shows a <delivered> element containing receipt message text as described in the "Receiving a Message Receipt" short_message table.
<send-fail>	If the message was validated and accepted by SMS Gateway but refused by the carrier, the receipt shows a <send-fail> element containing receipt message text as described in the "Receiving a Message Receipt" short_message table. Note that a send-fail receipt supersedes the SENT status given in the Job Status report, which indicates only that SMS Gateway successfully passed the message to the carrier.
<expired>	If the message's validity period has expired (default 36 hours) before the message could be delivered the receipt shows an <expired> element containing receipt message text as described in the "Receiving a Message Receipt" short_message table.
<send-notify>	If the message was accepted into the carrier network and DELIVER_NOTIFY was passed as the receipt option for the message a <send-notify> element containing receipt message text as described in "Receiving a Message Receipt" short_message table.

***NOTE**: By default, the report always shows send-fail receipts. Whether or not it also includes delivered, receipts depends on the Receipt Options you set in the Job Request XML document.*

Example Job Receipt Report

```
<job-receipt-report>
 <job-request-id>abc234354659234</job-request-id>
 <mqube-id>08urnjq00g003v0bk246419epvlk</mqube-id>
 <receipts>
  <receipt>
   <id>1pr4h5100g003v0bi244410d3p50</id>
   <msgid>11o0afq00g003v0bi244410d3nvg</msgid>
   <send-fail>id:314shi032gmheo2dcqq076dr8dnk sub:001 dlvrd:001 submit
date:1208260232 done date:1208260232 stat:UNDELIV err:00a text:This is some MT
text idd:null</send-fail>
  </receipt>
  <receipt>
    <id>29bl722061ghkv2f8sb5b4c9a3u4</id>
    <msgid>3h7q60s0f8eh9n2hh9mi45adg6l1</msgid>
    <send-notify>id:3h7q60s0f8eh9n2hh9mi45adg6l1 sub:001 dlvrd:001 submit
date:1208260238 done date:1208260238 stat:ENROUTE err:000
text:12345678901234567890 idd:REWA72MA_ES^15502555794039</send-notify>
  </receipt>
  <receipt>
    <id>3mkv78i2o88grf2ses1kt7553pla</id>
    <msgid>3h7q60s0f8eh9n2hh9mi45adg6l1</msgid>
    <delivered>id:15502555794039 sub:001 dlvrd:001 submit date:1208251939 done
date:1208251939 stat:DELIVRD err:000 Text:12345678901234567890</delivered>
  </receipt>
  <receipt>
    <id>3ajhm5u2angi6f2sf0tpl7sa0boi</id>
    <msgid>1hg2ldp2lvuhoj263gitu7109p6c</msgid>
    <expired>id:0705393711 sub:001 dlvrd:000 submit date:1208191903 done
date:1208221903 stat:EXPIRED err:017 text:abcdefghijklmnopqrst</expired>
  </receipt>
 <receipts>
</job-receipt-report>
```

3.3.3 JOB NOTIFICATION

When a job has finished, the SMS Gateway sends the following information via HTTP POST to the notification URL you specified in the Job Request:

```
JobRequestID=id
mQubeID=id
StatusURL=url
ReceiptURL=url
```

The data is URL encoded. For example:

```
JobRequestID=abc234354659234&mQubeID=08urnjq00g003v0bk246419epvlk&StatusU
RL=https%3A%2F%2Fstatus.mobilemessenger.com%2Fstatus%2Fgws%2F7fdhts45y
434908ksl78m21d8641%2FSMS%2F2007052218%2F08urnjq00g003v0bk246419epvlk-abc
234354659234.xml&ReceiptURL=https%3A%2F%2Fstatus.mobilemessenger.com%2
Fstatus%2Fgws%2F7fdhts45y434908ksl78m21d8641%2FSMS%2F2007052218%2F08urnjq
00g003v0bk246419epvlk-abc234354659234-receipts.xml
```

It's your responsibility to have a program running at the specified URL that will send back an HTTP 200 response .If SMS Gateway does not receive an appropriate response it will attempt to resend the notification every fifteen minutes for 8 hours. After that time, you'll need to request a manual resend from mGage.

3.4 Sending Individual MT Messages

As an alternative to constructing a multiple-recipient bulk send job, you can use the `sendSingle()` Web service to send individual MT messages. The message work flow is similar in both cases, since `sendSingle()` generates the same submit response and status reports as a bulk send job.

`sendSingle()` is callable via an HTTP POST/GET operation or through SOAP. Each method is described in detail below.

3.4.1 SENDSINGLE HTTP PARAMETERS

For the HTTP implementation, `sendSingle()` accepts the following input parameters.

HTTP	Description
`destination`	The recipient's mobile number, formatted in one of the following ways: • a 10-digit number (for example, `5008885555`) SMS Gateway will treat this as a North American Numbering Plan (NANP) number destined for a US or Canadian carrier. • an 11-digit number that starts with 1 (for example, `15008885555`) SMS Gateway will treat this as a NANP number destined for a US or Canadian carrier. • a variable-length number prefixed with + (for example, `+447700888555`) SMS Gateway will treat this as an international number that includes a country code prefix. See Appendix D: International Messages for additional information and a list of supported countries. Note that you can optionally include a `tel:` UMDA prefix for compatibility with earlier versions of SMS Gateway. For example: `tel:5008885555`
`message`	The text message in URL-encoded format. For example: `Hello+Wireless+World`
`serviceCode`	The short code you want to use for this message.

`productCode`	*mGage-specific parameter. Required for all MT messages.* A text string that defines the program or product to which the message content belongs. Product codes are set up by mGage to meet your specific needs. The format of the product code typically indicates the content provider, short code, item name, pricing, and program category. For example: ACMEX_12345_RINGHIP_000_D
`notificationURL` (optional)	The URL to which SMS Gateway should send the completion notification. For example: `http://mysite.com/sms/jobComplete.jsp` See Job Notification for more information.

`receiptOption` (optional)	**CRITICAL NOTE: Content Providers with subscription programs must use delivery receipts for the carriers that do not support deactivation files. This is the only way to detect a recycled (deactivated) MDN and cancel all subscriptions. Please contact your Account Manager for the list of carriers that do not support deactivation files.**
	
	The type of receipt that you want to appear on the Job Receipt Report. The following options are available:
	
	`SEND_FAIL`
	This notification is only sent when carrier side rejects the submission of the message. This is the default receipt type for the Job Receipt Report and will appear even if you omit the `<receipt-options>` parameter.
	`DELIVERED`
	This notification is supported for designated carriers only. This notification is sent when a handset Delivery Receipt is sent by the consumer's device acknowledging it has received the message.
	`DELIVER_NOTIFY`
	This notification is a combination of the two receipt options above with an additional case. It is sent when:
	• The carrier accepts or rejects the submission of the message.
	• And if the submission to the carrier is successful, a following handset delivery notification will be sent for supporting carriers only.
	
	See Job Receipt Report for more information.

`action`	*mGage-specific parameter. Required for all MT messages.* A text string that indicates the type and purpose of the message. Please refer to the Actions section for the available values for this parameter. You can optionally place an `<action>` tag inside individual `<r>` tags to override the default.

wapURL (optional)	If the message is a WAP push, set this parameter to the URL you want to appear on the handset. For example: `http://mysite.com?id=123&item=abc` The string you provide as the `message` parameter will become the title of the message that appears on the handset. The maximum length of the WAP URL and the message string combined shouldn't exceed 254 bytes, although the exact character count may vary slightly due to the WAP coding algorithm. For WAP pushes, you also need to set the `action` parameter (see above) to `WAPPUSH`. Note that some carriers have additional restrictions on WAP pushes. See Appendix C: Carrier-Specific Notes for more information.

3.4.2 CALLING SENDSINGLE VIA HTTP

To call `sendSingle()` via HTTP, send a POST to the following URL:

```
https://sendsms.mobilemessenger.com/wsgw/sendSingle
```

Pass the input parameters within the body of the HTTP request as a URL-encoded string. For example:

```
destination=5008885555&serviceCode=12345&message=A+giraffe
+goes+into+a+bar&productCode=ACMEX_12345_JOKE_199_S&action=CONTENT&notificatio
nURL=
http%3A%2F%2Fmysite.com%2Fsms%2FjobComplete.jsp&receiptOption=DELIVERED
```

Note that you also need to pass your mGage user ID and password for basic authentication.

For interactively testing in a browser, you can also call `sendSingle()` with an HTTP GET by passing the input parameters appended to the URL. For example:

```
https://sendsms.mobilemessenger.com/wsgw/sendSingle?&destination=
5008885555&serviceCode=12345&message=A+giraffe+goes+into+a+bar&
productCode=ACMEX_12345_JOKE_199_S&action=CONTENT&notificationURL=http%3A%2F%2
F
mysite.com%2Fsms%2FjobComplete.jsp&receiptOption=DELIVERED
```

3.4.3 SENDSINGLE HTTP RESPONSE

If the `sendSingle()` call is successful, the body of the HTTP response contains a unique 28-character message GUID (Globally Unique Identifier), confirming that SMS Gateway has accepted the message for processing. It also lists the URLs where you can view the Status Report and Receipt Report. For example:

```
Message Id: 1j0j9u0002bres006s43i3iu9mi0
StatusURL: https://status.mobilemessenger.com/status/gws/7fdhts45y43
4908ksl78m21d8641/SMS/2007052218/0u4v16j01g20890fgq466094jjcv-00q5djd01
g20890fgq466094jj80.xml
ReceiptURL: https://status.mobilemessenger.com/status/gws/7fdhts45y4
```

```
34908ksl78m21d8641/SMS/2007052218/0u4v16j01g20890fgq466094jjcv-00q5djd0
1g20890fgq466094jj80-receipts.xml
```

If the `sendSingle()` call fails, the HTTP response header contains the relevant error code. For example:

```
HTTP Status 400 - 11104- Missing value for destination
```

See sendSingle Error Codes for a list of possible errors.

The two critical pieces in the response are:

- The **StatusURL** in the response denotes to Job Status Report which shows the submission status of the message into mGage system.
- The **ReceiptURL** in the response denoted to Job Receipt Report which shows the submission status of the message into carrier system.

3.4.4 SENDSINGLE SOAP PARAMETERS

For the SOAP implementation, `sendSingle()` accepts the following input parameters.

SOAP	Description
`<aDestination>`	See the `destination` parameter in the sendSingle HTTP Parameters section above.
`<aMessage>`	See the `message` parameter in the sendSingle HTTP Parameters section above.
`<aServiceCode>`	See the `serviceCode` parameter in the sendSingle HTTP Parameters section above.
`<aProductCode>`	See the `productCode` parameter in the sendSingle HTTP Parameters section above.
`<aNotificationURL>` (optional)	See the `notificationURL` parameter in the sendSingle HTTP Parameters section above.
`<aReceiptOption>` (optional)	See the `receiptOption` parameter in the sendSingle HTTP Parameters section above.
`<aWapURL>` (optional)	See the `wapURL` parameter in the sendSingle HTTP Parameters section above.

<aExtraParams>	Additional parameters in the format *name=value*. The following parameters are currently supported: action (this is a mandatory parameter) optinType initialOptinTS secondaryOptinTS optinPin optinURL transactionId reverseAmount reverseReasonCode reverseNotes See the sendSingle HTTP Parameters section above for complete definitions of these parameters.

3.4.5 CALLING SENDSINGLE VIA SOAP

To call sendSingle() via SOAP, construct your SOAP XML with the input parameters contained inside a <sendSingle> operation tag. Be sure to use standard XML escape syntax for any special characters (&, <, “, ‘) in the XML data.

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soap-env:Body>
  <sendSingle>
   <aDestination>tel:6175551000</aDestination>
   <aMessage>Giraffe Joke</aMessage>
   <aServiceCode>12345</aServiceCode>
   <aProductCode>ACMEX_12345_JOKE_000_D</aProductCode>
   <aReceiptOption>DELIVERED</aReceiptOption>
   <aNotificationURL>http://mysite.com/jobComplete.jsp</aNotificationURL>
   <aWapURL>http://mysite.com?id=123&amp;item=abc</aWapURL>
   <aExtraParams>action=WAPPUSH</aExtraParams>
  </sendSingle>
 </soap-env:Body>
</soap-env:Envelope>
```

Post the XML to the following Web service endpoint:

https://ws.mobilemessenger.com/wsgw/services/MessagingWS

Note that you also need to pass your mGage user ID and password for basic authentication.

If you want to examine the sendSingle() interface or generate stub files for your applications, a corresponding WSDL file is available here:

https://ws.mobilemessenger.com/wsgw/services/MessagingWS?WSDL

3.4.6 SENDSINGLE SOAP RESPONSE

If the `sendSingle()` call is successful, the Web service returns a SOAP response containing the following elements:

Element	Description
`<messageId>`	A unique 28-character message GUID (Globally Unique Identifier), confirming that SMS Gateway has accepted the message for processing
`<reasonCode>`	The `<reasonCode>` element returns 0 if SMS Gateway accepted the message for processing. Otherwise it contains an error code. See sendSingle Error Codes for a list of possible causes.
`<reasonText>`	A description of the `reasonCode` error, if applicable.
`<receiptUrl>`	The URL where the Job Receipt Report is available. This report shows the submission status of the message into mGage system. The URL is a combination of your mGage organization ID, a time stamp, and two randomly-generated 28-character IDs: `https://status.mobilemessenger.com/status/gws/<orgID>/SMS/<timestamp>/<id1>-<id2>-receipts.xml`
`<statusUrl>`	The URL where the Job Status Report is available. This report shows the submission status of the message into carrier system. The URL is a combination of your mGage organization ID, a time stamp, and two randomly-generated 28-character IDs: `https://status.mobilemessenger.com/status/gws/<orgID>/SMS/<timestamp>/<id1>-<id2>.xml`

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Body>
  <sendSingleResponse
   soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <sendSingleReturn href="#id0"/>
  </sendSingleResponse>
  <multiRef id="id0" soapenc:root="0"
   soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
   xsi:type="ns1:SendSingleResult"
   xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
   xmlns:ns1="urn:MessagingWS">
   <messageId xsi:type="xsd:string">3ka3ioh2jliin52deil826vdmmlh</messageId>
   <reasonCode xsi:type="xsd:string">0</reasonCode>
```

```
   <reasonText xsi:type="xsd:string" xsi:nil="true"/>
   <receiptUrl xsi:type="xsd:string">https://status.mobilemessenger.com/
    status/gws/7fdhts45y434908ksl78m21d8641/SMS/
    2007052418/2jkOma63me6h1g25jbmq75po215e-1i1g2av3jpqgrd2kb
    n36e7j6ib24-receipts.xml</receiptUrl>
   <statusUrl xsi:type="xsd:string">https://status.mobilemessenger.com/
    status/gws/7fdhts45y434908ksl78m21d8641/SMS/
    2007052418/2jkOma63me6h1g25jbmq75po215e-1i1g2av3jpqgrd2kb
    n36e7j6ib24.xml</statusUrl>
  </multiRef>
 </soapenv:Body>
</soapenv:Envelope>
```

3.4.7 SENDSINGLE ERROR CODES

If the `sendSingle()` call fails, the Web service returns an error code and brief description. The following table lists the most common error conditions.

Error	Cause
03573 Destination phone number not found.	The mobile number is not routable.
11100 “xxx” is not a recognized UMDA transport	The mobile number contains a UMDA prefix other than “tel:”.
11103 Invalid format	One or more parameters has invalid contents. The `<reasonText>` element in the SOAP response (for SOAP calls) or the HTTP response header (for HTTP calls) contains details about the error.
11104 Missing value	One or more required parameters is empty. The `<reasonText>` element in the SOAP response (for SOAP calls) or the HTTP response header (for HTTP calls) contains details about the error.
15007 Invalid URL specified for Notification URL	The notification URL isn’t correctly formatted.
15010 Invalid receiptOption	The receipt option is invalid. The `<reasonText>` element in the SOAP response (for SOAP calls) or the HTTP response header (for HTTP calls) contains details about the error.
20700 Destination phone number not active	The mobile number has been deactivated.

3.5 Checking a Mobile Number

The `checkMobileNumber()` Web service method allows you to look up the wireless carrier and (optionally) the device model for a given mobile number. `checkMobileNumber()` is callable via HTTP POST/GET operations or through SOAP. Each method is described in detail below.

3.5.1 CALLING CHECKMOBILENUMBER VIA HTTP

For the HTTP implementation, `checkMobileNumber()` accepts the following input parameters:

Parameter	Description
`mobileNumber`	The subscriber's mobile phone number, with no spaces or non-numeric characters. For example: `5008885555` Note that you can optionally include a `tel:` UMDA prefix for compatibility with earlier versions of `checkMobileNumber()`. For example: `tel:5008885555`
`version` (optional)	Set to 2 (version=2) to get more detailed information regarding the subscriber's mobile number. Example: Subscriber mobile number is premium blocked / suspended / blocked. **Using version=2 is strongly recommended.**
`lookupDevice` (optional)	Set to `true` to include the device model in the response. The default is `false`. **Note:** Device lookup is currently only available for limited carriers. For unsupported carriers, please omit the `lookupDevice` parameter. Please consult with your Account Manager for understanding which carriers support lookupDevice feature.

To call `checkMobileNumber()` via HTTP, send an POST request to the following URL:

`https://ws.mobilemessenger.com/wsgw/checkMobileNumber`

Pass the input parameters within the body of the HTTP request as a URL-encoded string. For example:

`mobileNumber=5008885555&version=2`

`mobileNumber=5008885555&lookupDevice=true`

Note that you also need provide a mGage user ID and password for basic authentication.

For interactively testing in a browser, you can also call `checkMobileNumber()` with an HTTP GET by passing the input parameters appended to the URL. For example:

`https://ws.mobilemessenger.com/wsgw/checkMobileNumber?mobileNumber=5008885555&version=2`

checkMobileNumber HTTP Response

If the `checkMobileNumber()` HTTP call is successful, the response is an XML document containing the following elements:

Element	Description
`<carrierId>`	The carrier ID for the specified mobile number. See Appendix A: Carrier Information for the full list of carrier IDs. If the number is not routable, the `<carrierID>` element returns `-1` and the `<error>` element is set to the relevant error code. Example: If the number has been deactivated, the `<carrierID>` element returns the ID of the carrier to which the number last belonged and the `<error>` element is set to 20700. (See Appendix B: SMS Gateway Error Codes for a complete list of error codes.)
`<error>`	Returns empty if the carrier ID was found. Otherwise, contains an error code. For example: `03573- Invalid or non deliverable mobileNumber: 5008885555` See checkMobileNumber Error Codes for a list of error conditions.

Examples

If the number lookup is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<checkMobileNumber>
 <carrierId>2</carrierId>
 <error/>
</checkMobileNumber>
```

If the number is not routable:

```
<?xml version="1.0" encoding="UTF-8"?>
<checkMobileNumber>
 <carrierId>-1</carrierId>
 <error>03573- Invalid or non deliverable mobileNumber: 5008885555</error>
</checkMobileNumber>
```

If the number is deactivated:

```
<?xml version="1.0" encoding="UTF-8"?>
<checkMobileNumber>
 <carrierId>2</carrierId>
 <error>20700- Phone number deactivated. Undeliverable mobileNumber:
5008885555.
```

```
 </error>
</checkMobileNumber>
```

If `lookupDevice=true`, the XML response also includes a `<device>` element containing the following additional device lookup elements:

Element	Description
`<model>`	The device model for the specified mobile number. For example: `MOTOROLA V266`
`<error>`	Returns empty if the device model was found. Otherwise contains an error code and associated description. For example: `15011- Device lookup not performed due to number lookup error` See checkMobileNumber Error Codes for a list of error conditions. **Note:** If the device lookup error wasn't caused by a problem with the mobile number itself, the `<carrierId>` element will still return the correct carrier code.

Examples

If the device lookup is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<checkMobileNumber>
 <carrierId>7</carrierId>
 <error/>
 <device>
  <model>MOTOROLA V266</model>
  <error/>
 </device>
</checkMobileNumber>
```

If data for this device isn't available:

```
<?xml version="1.0" encoding="UTF-8"?>
<checkMobileNumber>
 <carrierId>7</carrierId>
 <error/>
 <device>
  <model/>
  <error>
   15014- Carrier returned error: java.rmi.RemoteException: Could not found
model for mdn = 5008885555
  </error>
 </device>
</checkMobileNumber>
```

3.5.2 CALLING CHECKMOBILENUMBER VIA SOAP

The SOAP implementation provides two separate `checkMobileNumber()` methods:

Method	Description
`checkMobileNumber()`	Returns the carrier ID for the supplied mobile number.
`checkMobileNumberDevice()`	Returns the carrier ID and the device model for the supplied mobile number. **Note:** Device lookup is currently only available for Verizon Wireless. For other carriers, please use only the `checkMobileNumber()` method. Also, if you're querying an Verizon Wireless number but don't need the device lookup, `checkMobileNumber()` provides a faster response.
`checkMobileNumberV2()`	Returns the carrier ID and detailed information regarding the subscriber's mobile number. Example: Subscriber mobile number is Premium Blocked/ suspended/blocked.
`checkMobileNumberDeviceV2()`	Returns the carrier ID, the device model and detailed information for the supplied mobile number. **NOTE:** Device lookup is currently only available for limited carriers. For unsupported carriers, please omit the `lookupDevice` parameter. Please consult with your Account Manager for understanding which carriers support lookupDevice feature.

Both methods accept the following input parameter:

Parameter	Description
`<aMobileNumber>`	The subscriber's mobile phone number, with no spaces or non-numeric characters. For example: `5008885555` Note that you can optionally include a `tel:` UMDA prefix for compatibility with earlier versions of `checkMobileNumber()`. For example: `tel:5008885555`

To call checkMobileNumber(), construct your SOAP XML with the <aMobileNumber> input parameter contained inside a <checkMobileNumber> operation tag:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Body>
  <checkMobileNumber>
   <aMobileNumber>5008885000</aMobileNumber>
  </checkMobileNumber>
 </soapenv:Body>
</soapenv:Envelope>
```

To call checkMobileNumberDevice(), construct your SOAP XML with the <aMobileNumber> input parameter contained inside a <checkMobileNumberDevice> operation tag:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Body>
  <checkMobileNumberDevice>
   <aMobileNumber>5008885000</aMobileNumber>
  </checkMobileNumberDevice>
 </soapenv:Body>
</soapenv:Envelope>
```

Post the XML to the following Web service endpoint:

`https://ws.mobilemessenger.com/wsgw/services/MessagingWS`

Note that you also need to pass your mGage user ID and password for basic authentication.

If you want to examine the checkMobileNumber() or checkMobileNumberDevice() interfaces or generate stub files for your applications, a corresponding WSDL file is available here:

`https://ws.mobilemessenger.com/wsgw/services/MessagingWS?WSDL`

checkMobileNumber SOAP Response

If the checkMobileNumber() SOAP call is successful, the Web service returns a CheckMobileNumberResult response object containing the following string elements:

Element	Description
<carrierId>	The carrier ID for the specified mobile number. See Appendix A: Carrier Information for the full list of carrier IDs. If the number is not routable, the <carrierID> element returns -1 and the <reasonCode> element is set to the relevant error code. If the number has been deactivated, the <carrierID> element returns the ID of the carrier to which the number belonged and the <reasonCode> element is set to error 20700. (See Appendix B: SMS Gateway Error Codes for a complete list of error codes.)
<reasonCode>	The returns 0 if the carrier ID was found. Otherwise, contains an error code. See checkMobileNumber Error Codes for a list of error conditions. **Note:** The <reasonCode> element omits any leading zeros from the error code string. For example, 03573 is rendered as 3573.
<reasonText>	A description of the <reasonCode> error, if applicable. For example: 03573- Invalid or non deliverable mobileNumber: 5008885555

Examples

If the number lookup is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Body>
  <checkMobileNumberResponse
   soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <checkMobileNumberReturn href="#id0"/>
  </checkMobileNumberResponse>
  <multiRef id="id0" soapenc:root="0"
   soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
   xsi:type="ns1:CheckMobileNumberResult"
   xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
   xmlns:ns1="urn:MessagingWS">
   <carrierId xsi:type="soapenc:string">2</carrierId>
   <reasonCode xsi:type="soapenc:string">0</reasonCode>
   <reasonText xsi:type="soapenc:string" xsi:nil="true"/>
  </multiRef>
 </soapenv:Body>
</soapenv:Envelope>
```

If the number is not routable:

```
...
   <carrierId xsi:type="soapenc:string">-1</carrierId>
   <reasonCode xsi:type="soapenc:string">3573</reasonCode>
   <reasonText xsi:type="soapenc:string">03573- Invalid or non
deliverable mobileNumber: 5008885000</reasonText>
...
```

If the number is deactivated:

```
...
   <carrierId xsi:type="soapenc:string">2</carrierId>
   <reasonCode xsi:type="soapenc:string">20700</reasonCode>
   <reasonText xsi:type="soapenc:string">20700- Phone number deactivated.
Undeliverable mobileNumber: 5008885555</reasonText>
...
```

checkMobileNumberDevice SOAP Response

If the `checkMobileNumberDevice()` SOAP call is successful, the Web service returns a `CheckMobileNumberResult` response object containing the same elements as the `checkMobileNumber()` call (see above). In addition, it returns a `DeviceLookupResult` response object containing the following string elements:

Element	Description
`<model>`	The device model for the specified mobile number. For example: `MOTOROLA V266`
`<reasonCode>`	Returns 0 if the device model was found. Otherwise contains an error code. See checkMobileNumber Error Codes for a list of error conditions. **Note:** If the device lookup error wasn't caused by a problem with the mobile number itself, the `<carrierId>` element will still return the correct carrier code.
`<reasonText>`	A description of the `<reasonCode>` error, if applicable. For example: `15011- Device lookup not performed due to number lookup error`

Examples

If the device lookup is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
```

```
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Body>
  <checkMobileNumberDeviceResponse
   soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <checkMobileNumberDeviceReturn href="#id0"/>
  </checkMobileNumberDeviceResponse>
  <multiRef id="id0" soapenc:root="0"
   soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
   xsi:type="ns1:CheckMobileNumberDeviceResult"
   xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
   xmlns:ns1="urn:MessagingWS">
   <lookupResult href="#id1"/>
   <result href="#id2"/>
  </multiRef>
  <multiRef id="id1" soapenc:root="0"
   soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
   xsi:type="ns2:DeviceLookupResult" xmlns:ns2="urn:MessagingWS"
   xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
   <model xsi:type="xsd:string">MOTOROLA V266</model>
   <reasonCode xsi:type="xsd:string">0</reasonCode>
   <reasonText xsi:type="xsd:string" xsi:nil="true"/>
  </multiRef>
  <multiRef id="id2" soapenc:root="0"
   soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
   xsi:type="ns3:CheckMobileNumberResult" xmlns:ns3="urn:MessagingWS"
   xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
   <carrierId xsi:type="xsd:string">7</carrierId>
   <reasonCode xsi:type="xsd:string">0</reasonCode>
   <reasonText xsi:type="xsd:string" xsi:nil="true"/>
  </multiRef>
 </soapenv:Body>
</soapenv:Envelope>
```

If data for this device isn't available:

```
...
   <model xsi:type="soapenc:string" xsi:nil="true"/>
   <reasonCode xsi:type="soapenc:string">15014</reasonCode>
   <reasonText xsi:type="soapenc:string">15014- Carrier returned error:
java.rmi.RemoteException: Could not found model for mdn = 5008885555
   </reasonText>
...
```

3.5.3 CHECKMOBILENUMBER ERROR CODES

If the `checkMobileNumber()` call fails, the Web service returns an error code and brief description. The following table lists the most common error conditions.

Error	Description	Retriable?
03573 Invalid or non deliverable mobileNumber	The mobile number is invalid or not routable.	No
03577 Service currently unavailable	The database or LERG server is temporarily unavailable.	Yes
11100 “xxx” is not a recognized UMDA transport	The mobile number contains a UMDA prefix other than `tel:`.	No
11103 Invalid format for Mobile number	The mobile number is incorrectly formatted.	No
11104 Missing value for Mobile number	No mobile number was specified.	No
20700 Phone number deactivated. Undeliverable mobileNumber	The mobile number has been deactivated.	No
20701 MDN is suspended	The destination MDN is suspended by the carrier due to unpaid phone bill. Do not send MT to this MDN until it gets unsuspended.	No
20712 MDN is blocked	The destination MDN is blocked for all 3rd party messaging services.	No
20713 MDN is premium blocked	The destination MDN is blocked for premium programs. SMS messaging should work for this MDN.	No
20714 MDN is ineligible	The destination MDN is a reseller of the carrier and does not support billing. SMS messaging should work for this MDN.	No
20715 MDN is barred	The destination MDN is barred in mGage system.	No

The device lookup functionality may return the following additional errors:

Error	Description	Retriable?
15011 Device lookup not performed due to number lookup error	Device lookup couldn't take place due to a problem with the mobile number. See the base `checkMobileNumber()` errors above for details.	Depends on reason
15012 Device lookup is not supported by this carrier	The carrier does not support device lookups.	No
15013 Lookup service unavailable	The device lookup service is unavailable at the carrier or the connection to the carrier is down.	Yes
15014 Carrier returned error	The carrier's lookup service returned an error. The details are included in the `<error>` element (HTTP) or the `<reasonText>` element (SOAP).	Depends on reason
15015 Device lookup is busy. Try again later	SMS Gateway has reached the maximum allowed number of concurrent lookup requests. Since device lookups require SMS Gateway to query an external carrier-maintained database whose performance may vary, mGage limits the number of concurrent requests in order to protect overall server performance. If find that you're routinely triggering error 15015, please try throttling back the rate at which your application makes device lookup requests.	Yes

3.5.4 SENDING WAP PUSH REQUESTS

You can send WAP Push requests via the Web Services interface by setting the `wapURL` parameter in your `sendSingle()` call. See Sending Individual MT Messages for more information.

3.6 Receiving MO Messages

For the Web Services interface, SMS Gateway delivers MO messages via an HTTP POST to a provided URL. The body of the POST contains the following parameters as URL-encoded key-value pairs.

Parameter	Description
`sender`	The mobile phone number from which the message originated.
`serviceCode`	The service code used for the message.
`messageId`	The mGage message identifier.
`content`	The text of the message.
`carrierId`	The ID of the carrier from which the message was sent. See Appendix A: Carrier Information for the full list of carrier IDs.
`senderCity`	The sender's city information (if sent by the carrier).
`senderState`	The sender's state information (if sent by the carrier).
`dialog.*`	Deprecated parameters included for compatibility with earlier versions of the API.

concatId concatTotalParts concatPart	**Multipart MO messages** mGage is able to receive concatenated SMS-MO messages from the carriers. By default, only the first partial SMS-MO is sent to the Content Provider, for MOs that are multipart. Multipart MOs can be enabled for Content Providers, please contact your Account Manager. concatId: The reference number for a particular concatenated MO. This is constant across all parts of the concatenated MO. concatTotalParts: Indicates the total number of MOs within the concatenated MO. concatPart: Indicates the sequence number of this MO within the concatenated MO. concatId and concatTotalParts values remain constant for a multipart MO message, however, concatPart keeps incrementing for following parts of the message. For example: 1st part of multipart messsage: concatId=59, concatTotalParts=2, and **concatPart=1** 2nd part of multipart messsage: concatId=59, concatTotalParts=2, and **concatPart=2** Please see example below.

Example MO:

```
content-type: application/x-www-form-urlencoded; charset=UTF-8

user-agent: Jakarta Commons-HttpClient/3.0

host: bastion.m-qube.com:8080

content-length: 593

concatId=59&concatTotalParts=2&dialog.service_code=37717&dialog.carrier_id=5&sender=3372124786&content=788896+Loneliness+Because+Edmund+wasn%27t+very+lonely+when+he+was+by+himself.+Sometimes+he+was+but+it+wasn%27t+the+main+ideal.+After+Edmund+betrayals+his+bro&carrierId=5&senderCity=NotAvailable&concatPart=1&serviceCode=37717&senderState=NotAvailable&dialog.umda=tel%3A3372124786&messageId=2l4cne21fegiaq2ho3heh6tlslr1&dialog.message=788896+Loneliness+Because+Edmund
```

```
+wasn%27t+very+lonely+when+he+was+by+himself.+Sometimes+he+was+but+it+wasn%27t
+the+main+ideal.+After+Edmund+betrayals+his+bro
```

It's your responsibility to have a server program running at the specified URL that will accept the POST with a standard HTTP 200 response. If a receiving program isn't present, or if the POST fails for any reason, SMS Gateway will retry delivery ten times over a period of 15 minutes. After ten failed delivery attempts, the messages are placed in a dead letter queue. Since the delivery failures trigger an alarm in SMS Gateway, you will be contacted by mGage Support if the problem appears to lie with your server. Once the server error is fixed, mGage can resend the queued messages.

Please contact your mGage Account Manager for more information about receiving MO messages and to set up an appropriate receiving URL.

3.7 Receiving Delivery Receipts

By default the Web Services interface will write DRs into an XML receipt document. If desired, DRs can instead be pushed via an HTTP POST to a provided URL. The body of the POST contains the following parameters as URL-encoded key-value pairs.

The features, receiving pushed Delivery Receipts (explained here) and using the Job Receipt Report (default method) are mutually exclusive, so when push Delivery Receipts are turned on, the receipts URL contains no data.

***CRITICAL NOTE**: Content Providers with subscription programs must use delivery receipts for the carriers that do not support deactivation files. This is the only way to detect a recycled (deactivated) MDN and cancel all subscriptions. Please contact your Account Manager for the list of carriers that do not support deactivation files.*

Parameter	Description
`mdn`	The mobile phone number of the associated MT. (Note there is currently an issue where the format of the phone number is not consistent. Please lookup the phone number from the associated MT until this is fixed.)
`serviceCode`	The service code of the associated MT.
`mtMsgId`	The mGage message identifier of the associated MT.
`drMsgId`	The mGage message identifier of the DR.
`content`	The text of the DR. See "Receiving a Message Receipt" short_message table for information on the format
`carrierId`	The ID of the carrier from which the message was sent. See Appendix A: Carrier Information for the full list of carrier IDs.
`messageType`	Always will be set to "DR"

deliveryState	The delivery state of the associated MT. [ENROUTE|DELIVERED| FAIL|REJECTED|EXPIRED|UNDELIV|UNKNOWN]

It's your responsibility to have a server program running at the specified URL that will accept the POST with a standard HTTP 200 response. If a receiving program isn't present, or if the POST fails for any reason, SMS Gateway will retry delivery once an hour for up to 24 hours. Since the delivery failures trigger an alarm in SMS Gateway, you will be contacted by mGage Support if the problem appears to lie with your server. Once the server error is fixed, mGage can resend the queued messages.

Please contact your mGage Account Manager for more information about receiving DR messages and to set up an appropriate receiving URL.

Example DR POST

```
content-type: application/x-www-form-urlencoded; charset=UTF-8

user-agent: Jakarta Commons-HttpClient/3.0

host: bastion:8080

content-length: 314

carrierId=2&content=id%3A188ot6j0sg84lc0dou4rs02tjifv+sub%3A001+dlvrd%3A001+submit+date%3A1206071913+done+date%3A7001010000+stat%3ADELIVRD+err%3A000+text%3Ac+idd%3Anull&deliveryState=DELIVERED&drMsgId=2elefl2134uhmo2rn31ii4fq0nk1&mdn=9999299112&messageType=DR&mtMsgId=3m3aoh7360kgf2219tb5h5bej10d&serviceCode=57758
```

APPENDIX A: CARRIER INFORMATION

The full list of supported carriers for US and Canada Market is available in the below Google sheet. This document is updated periodically as we add more carriers and feature support.

http://goo.gl/ar90hg

For International carrier support and features, please talk to your Account Manager.

APPENDIX B: SMS GATEWAY ERROR CODES

Please contact your Account Manager for latest Error Code document.

APPENDIX C: CARRIER-SPECIFIC NOTES

Sprint WAP Push URLs

Sprint only allows its users to download content from preapproved URLs. If you attempt to send a WAP push that uses a non-recognized location, the download is blocked and the recipients see an "Untrusted Content Error" on their handset. In order to avoid this error, you'll need to ask Sprint to set up a whitelisted DNS alias that maps to your actual content URL. If you need help with this process, please contact the mGage Connectivity Team.

APPENDIX D: INTERNATIONAL MESSAGES

Character Encoding

To enable delivery to international numbers, SMS Gateway accepts messages in several different character sets. It then transcodes the messages into the character set most appropriate or the target country. Any characters that aren't available in the final encoding scheme won't render correctly on the handset, so please ensure that your international messages use only characters appropriate to the destination country.

The currently-supported character sets are as follows:

Character Set	Maximum Message Length
Latin-1	140 characters (160 characters if all are GSM-compatible *)
UTF-8	140 characters (160 characters if all are GSM-compatible *)
GSM	160 characters
UCS-2	70 characters

* GSM is optimized for European languages and uses a 7-bit packing scheme to provide up to 20 additional characters per message. If all the characters in your Latin-1 or UTF-8 message are compatible with the GSM 7-bit default alphabet, SMS Gateway automatically transcodes to GSM before sending. However, note that some GSM characters (such as the Euro symbol €) consume two bytes, reducing the available character count accordingly.

Over-length messages are either rejected or truncated depending on how your connection is configured. Contact the mGage Connectivity Team for more information.

Default Inbound Encoding

For the Web Service interface, SMS Gateway expects all inbound messages to be UTF-8.

For the SMPP interface, SMS Gateway expects inbound messages to use the default encoding defined when mGage set up your bind. However, you can submit individual messages that use different encoding by setting the `data_coding` parameter in the `submit_sm` PDU:

Character Set	data_coding Value
Latin-1	0x03
UTF-8	Must be set as the default
GSM	Must be set as the default
UCS-2	0x08

Country Codes and Routes

For the list of country supported and relevant encoding schemes, please contact account manager or mGage technical support team.

APPENDIX E: ACTIONS

Here are the possible values for the “action” parameter that is required to be sent by Content Providers for each message along with IPC (product_code).

Action	Action Description
CONTENT	Content message. Examples: "XYZ Alerts: MLB World Series winner is..." “Bank: Your one time PIN is...".
HELP	CP sends this as a reaction to all HELP-MO. *Example:* *"XYZ Alerts: This is a service that sends daily news alerts...".*
SYSTEM	CP sends this for system related or other message types (i.e. consumer sends an MO with an unknown keyword) *Example:* *"XYZ Alerts: You have send an unknown keyword".*
WAPPUSH	CPs should use this action when the MT is a WAP-Push SMS message. There are other parameters to set and carrier restrictions for WAP-Push messaging, please refer to WAP-Push information in this document.