Dynmark HTTP Interface Technical Specification

Revision 3.0.3

24 June 2005

E-mail: support@dynmark.com

Contents

1 Introduction

1.1 Overview

This document describes Dynmark's HTTP interface for sending and receiving SMS messages to and from mobile handsets via the Dynmark Messaging Platform (DMP).

Messages are specified using a simple XML document format that is transmitted between the customer and the DMP using the HTTP protocol. Familiarity with the references given in Appendix A is assumed. In particular, customers must have a full understanding of the XML and HTTP protocols specified in references [XML] and [HTTP] respectively.

Use of the DMP is subject to Dynmark's Terms and Conditions which are available at http://www.dynmark.com, by sending an e-mail to support@dynmark.com, or by written request to:

Al Shortland
Dynmark International Limited
Church Mews
Church Street
Cheltenham
Gloucs
GL50 3HA
+44 (0)1242 257989

1.2 Definitions

The definitions listed in Table 1 are used throughout this document.

Table 1: Definitions

Account	A customer account providing access to the DMP.
Client Application	A customer application which uses the HTTP interface described in this document.
Concatenated SMS Message	A binary message consisting of more than one SMS segment. Under [GSM0340], multiple concatenated SMS segments must be used to send a binary SMS message where the length of the SMS message exceeds 140 bytes.
Customer	A DMP user with an authorised account.
DMP	The Dynmark Messaging Platform.
Downstream Provider	A Dynmark partner organisation involved in transmitting SMS messages.
Handset	A mobile telephone or other device capable of receiving and displaying SMS messages.
HTTP Interface	The XML/HTTP interface to the DMP described in this document.
Message	A message sent using the Short Message Service.
Message Submission	The HTTP interface request used to send SMS messages through the DMP.
Message Template	An HTTP interface XML element used to describe SMS messages. A message template may be used to send a message with common characteristics to multiple handsets.
Method	A DMP procedure invoked by submission of a defined XML document format in an HTTP request.
MO Message	A Mobile Terminated SMS message; that is, an SMS message sent by a handset.
MT Message	A Mobile Terminated SMS message; that is, an SMS message destined for a handset.
Operator	The mobile network operator used by the handset at the time of

	SMS message receipt.
Request	An XML document submitted by the customer to the DMP using the XML/HTTP interface.
Response	An XML document returned by the DMP to the customer in response to an XML/HTTP interface request.
SMS	The Short Message Service, as defined by [GSM0340].
SMS Segment	A single binary SMS message constituting part of a Concatenated SMS Message.
Text Message	An SMS text message containing a subset of the limited characters specified in the GMS default alphabet, as defined in see [GSM0338]. The maximum length of a 7-bit text message is 160 characters.

1.3 Prerequisites

Before using the HTTP interface, a customer must have at least one authorised DMP account. Customer accounts are discussed further in Section 2.2.

For use of the programmatic interface, the customer must also have a client application that conforms to this specification.

2 Message Sending Interface Specification

2.1 Overview

Customers send MT messages via the DMP by submitting an XML document in an HTTP POST request. The DMP responds by returning an HTTP response to the customer containing a second XML document. This section provides all the details necessary to use this interface from a client application.

The HTTP interface uses an XML Schema document to validate the submitted XML request. The schema document may be viewed at the URL given in Table 2. Xml Schema is defined by references [XMLS].

Table 2: URL of XML Schema used for validation of message sending request

http://80.82.139.132/XmlHttp/MessageMTSend.xsd

Customers may manually test the interface by pointing a browser at the URL given in Table 3.

Table 3: Request URL for manual testing of message sending

http://80.82.139.132/SmsPress/Rpc/etxtmessagemtsend.htm

The form located at this URL shows the skeleton XML document format used to submit messages to the DMP. By completing the XML in accordance with the specifications given in this section, the user may manually send messages and view the interface response. Note that this manual version of the interface is intended for test purposes only. A client application should be employed for production use of the interface.

In the following sections, all attribute types are those defined in the XML Schema specifications [XMLS].

2.2 Customer Accounts

2.2.1 Overview

Each customer wishing to send MT messages using the DMP must have at least one authorised DMP message MT sending account. An account provides access to the DMP HHTP message sending interface by means of a unique account name and a corresponding password. Dynmark may configure each account with the parameters described in this section. Dynmark will notify each customer of their account settings when each account is set up. If you are unsure of your account settings, or if you wish to set up additional accounts, please contact your Dynmark Account Manager or send an e-mail to support@dynmark.com

Customer Accounts are independent of any Customer Gateway Accounts configured for MO message receiving.

Note that separate MT message sending accounts are required for sending of free MT messages, and for sending of premium MT messages to each of the supported mobile networks.

2.2.2 Access Restrictions

Each DMP customer account may be configured to restrict access by the IP address of the client application making the interface request. This measure is designed to protect both customers and Dynmark from unauthorised use of accounts.

Restricted accounts must specify at least one IP address or IP address range with permission to access the DMP. Customers may request the addition of IP addresses or ranges to their account access list by sending an e-mail to support@dynmark.com

2.2.3 Message Sending Limits

Each DMP customer account for MT message sending may be configured to restrict the number of messages that can be sent in a given timeframe (daily or monthly), or on the basis of message sending credits. These measures are designed to regulate account use, and to guard against account abuse by any unauthorised third parties.

Customers wishing to query their message sending limits should send an e-mail to support@dynmark.com

2.3 Message Request

2.3.1 Request Transmission

The client application initiates a request to send bulk MT messages by opening a connection to the HTTP interface server using the settings listed below.

Table 4: Server settings for message sending

Server Name	`80.82.139.132`
Server Port	`443`
Protocol	`TCP`

The client application must then transmit an HTTP POST request to the server using the HTTP settings listed below.

Table 5: HTTP POST settings for message sending

HTTP POST Request URL	`/SmsPress/etxt/MessageMTSend.aspx`
HTTP Content Type	`text/xml`
HTTP Content Length	*`<length of HTTP request body>`*

The HTTP POST request body must contain an XML document which conforms to the specification given in Section 2.3.3.

Many of the HTTP client libraries available on various platforms will transparently handle server connection and initiation of an HTTP POST request when supplied with a request URL. The URL for use with these libraries is given in the following table.

Table 6: Request URL for HTTP client library message sending

`http://80.82.139.132/SmsPress/etxt/MessageMTSend.aspx`

2.3.2 Example Request

Listing 1 shows an example of an XML request used to send bulk SMS messages with merged data fields to multiple handsets. The XML shown constitutes the body of the HTTP POST request.

Listing 1: XML document showing example of message sending request

```
<?xml version="1.0" encoding="utf-8"?>
<Dmp xmlns="http://80.82.139.132/XmlHttp/MessageMTSend.xsd" version="1.0">
<Account name='USERNAME' password='PASSWORD'/>
<MessageMTSend test="false" split="true" client_type="SMSPRESS"
MessageGUID="942497c4-ebfc-4f0c-81bc-8b39542ba34b" SendDeliveryStatus="true"
DeliveryStatusUrl="http://www.yourserver.com/etxt/receivemsg.asp">
<MessageMT type="Text.GSM0338" class="1" deliverAfter="2006-03-23T12:49:50">
<Originator>dynmarko</Originator>
<Content>Dear {MF2} {MF1} New field is {MF4}
Your zipcode is {MF3}.
Thanks</Content>
<MergeFieldOrder value="{MF1}~{MF2}~{MF3}~{MF4}" />
<Recipient userRef="1" o=""
mfv="oscar~jenkins~CJ2GH~fieldOJ#!#John~Hamilton~UB60PU~fieldJH#!#firstname~lastna
me~zipcode~newfield#!#"> <Number>4472341234,4482341234,447971893841,</Number>
</Recipient>
</MessageMT>
</MessageMTSend>
</Dmp>
```

The XML format shown in Listing 1 is described in detail in the following section. Note that the values for account name, account password and recipient number are for illustrative purposes only and would need to be replaced by actual values in a real application.

2.3.3 XML Document Format

2.3.3.1 XML Declaration

The `<?xml>` element is required to identify the document type to the server, and it must appear as the first line in each submission. No other characters must precede this element.

The required `version` attribute must have the value `'1.0'`, and the required `encoding` attribute must have the value `'utf-8'`.

Table 7: XML declaration

Element	**`<?xml>`**
Parent Element	-
Has Attributes	`version` `encoding`
Child Elements	-
Character Data	-
Attribute	**`version`**
Parent Element	`<?xml>`
Type	string
Usage	required
Values	must be `"1.0"`
Default Value	-
Occurrences	1
Attribute	**`encoding`**
Parent Element	`<?xml>`
Type	string
Usage	required
Values	must be `"utf-8"`
Default Value	-
Occurrences	1

2.3.3.2 Dmp Element

The required `<Dmp>` element is the document root element and represents a generic requst submission made to or from the DMP.

The required `xmlns` attribute supplies the document's default XML namespace to the server and is required for validation of the submission against the specified XML Schema (see references [XMLNS] and [XMLS]). This attribute must be present and have the stated value to ensure correct request processing.

The required `version` attribute is used to ensure compatibility between different versions of the DMP. At present, its value must be `'1.0'`.

Table 8: Dmp element

Element	**`<Dmp>`**
Parent Element	-

Child Elements	`<Account>`
	`<MessageMTSend>`

Has Attributes	`xmlns` `version`
Character Data	-
Usage	Required
Occurrences	1
Attribute	**`xmlns`**
Type	string
Usage	required
Values	must be 'http://80.82.139.132/XmlHttp/MessageMTSend.xsd'
Default Value	-
Attribute	**`version`**
Type	string
Usage	required
Values	must be '1.0'
Default Value	-

2.3.3.3 Account Element

The required `<Account>` element represents the customer's DMP account through which SMS messages are sent. The `name` and `password` attributes are both required and must specify the account credentials supplied by Dynmark.

Table 9: Account element

Element	**`<Account>`**
Parent Element	`<Dmp>`
Child Elements	-
Has Attributes	`name` `password`
Character Data	-
Usage	required
Occurrences	1
Attribute	**`name`**
Type	string
Usage	required
Values	*the name of a valid DMP account supplied by Dynmark and authorised for use by the customer*
Default Value	-
Attribute	**`password`**
Type	string
Usage	required
Values	*the Dynmark-supplied password of the account specified by the* `name` *attribute*
Default Value	-

2.3.3.4 MessageMTSend Element

The required <MessageMTSend> element represents the DMP method call used to send messages.

The optional test attribute indicates whether messages should actually be sent to the specified handsets.

test='false' indicates that messages should be fully processed by the DMP and delivered to the specified handsets.

test='true' indicates that the DMP should perform initial validation of the message send submission, but it should not deliver messages to the specified handsets, or treat the messages as delivered for billing or message limiting purposes. This setting is useful during the development of client applications.

Table 10: MessageMTSend element

Element	**<MessageMTSend>**
Parent Element	<Dmp>
Child Elements	<MessageMT>
Has Attributes	test
Character Data	-
Usage	Required
Occurrences	1
Attribute	**test**
Type	boolean
Usage	Optional (currently not in use)
Values	'true' 'false'
Default Value	'false'
Attribute	**split**
Type	boolean
Usage	optional
Values	'true' – If message length is more than 160 characters then the message will be split into 160 character segments & you will charged more message credits. 'false' – If message length after mailmerge becomes more than 160 character, then the message will be deleted.
Default Value	'false'
Attribute	**client_type**
Type	string
Usage	required
Values	”SMSPRESS” – do not change this
Default Value	none
Attribute	**MessageGUID**
Type	String
Usage	Optional. Helps to maintain a transaction. On occasion, DMP Server will accept your request but while it responds you could be disconnected. If you resend your request with same GUID then we will not run the transaction

	again and update you with status of that transaction.
Values	System generated GUID eg: "942497c4-ebfc-4f0c-81bc-8b39542ba34b"
Default Value	none
Attribute	**SendDeliveryStatus**
Type	boolean
Usage	Optional. Flag to indicate if delivery report is required.
Values	SendDeliveryStatus="true" or SendDeliveryStatus="false"
Default Value	none
Attribute	**DeliveryStatusUrl**
Type	String
Usage	Optional. The URL of a page on the customer server to receive the delivery report notification. It is up to the customer how the message is parsed and handled. To the supplied URL, the DMP returns the following query-string. For example: http://80.82.139.132/SmsPress/receivemsg2.asp?dmp_dynmarkref=56A8408CD4174AC2A0DC8A9330C4C5AC&messageguid=642497c4-ebfc-4f1c-81bc-8b39542ba34y&statusid=21&recipient=447841831013&statusdescription=Messages%20delivered%20to%20handset&datetime=2005-04-18T16:21:40.153 If within 8 seconds, the customer server does not return HTTP 200 status (success status) then we retry at 1, 2, 4, 8, 16, 32 minutes Data being sent is: dmp_dynmarkref=56A8408CD4174AC2A0DC8A9330C4C5AC - this is the unique identifier of this message on the DMP. messageguid=642497c4-ebfc-4f1c-81bc-8b39542ba34y - The unique GUID sent by customer during original MT message submission statusid=21 – The receipt status of the message. Various values are: 1 Messages delivered to handset 2 Messages not delivered during validity period 3 Messages queued for delivery at SMSC 4 Messages rejected by operator 5 Messages sent to discontinued/invalid number 6 Messages sent to handset with full SIM 7 Messages SMSC was not able to send 8 Messages to user with no credit 9 Messages with no response from Mobile Operator recipient=447841831013 - number to which message was sent statusdescription=Messages%20delivered%20to%20handset - same description as given in statusid notes (URLencoded string) datetime=2005-04-18T16:21:40.153 - The time of the receipt. Date format is yyyy-mm-ddThh:mm:ss:mmm with T as separator - this universal standard.
Values	DeliveryStatusUrl="http://www.yourserver.com/etxt/receivemsg.asp"
Default Value	none

2.3.3.5 MessageMT Element

The required <MessageMT> element represents a message template used to specify an SMS message to be sent.

The optional deliverAfter attribute specifies the date and time after which the DMP should attempt delivery of the messages specified in the message template. If this attribute is omitted, then delivery is attempted as soon as possible following submission.

Table 11: MessageMT element

Element	**<MessageMT>**
Parent Element	<MessageMTSend>
Child Elements	<Originator> <Content> <Recipient>
Has Attributes	type class deliverAfter
Character Data	-
Usage	Required
Occurrences	1 to unlimited
Attribute	**type**
Type	string
Usage	*optional*
Values	'Text.GSM0338'
Default Value	'Text.GSM0338'
Attribute	**class**
Type	nonNegativeInteger
Usage	optional
Values	'1'
Default Value	'1'
Attribute	**deliverAfter**
Type	string
Usage	*Optional - time at which you want the message to be delivered: YYYY-MM-DDTHH:MM:SS format*
Values	Eg: "2006-03-23T12:49:50"
Default Value	none

2.3.3.6 Originator Element

The required `<Originator>` element allows dynamic setting of the TP-Originating-Address (TP-OA) of the SMS messages described by the message template. The element contains only character data, which is used to set the TP-OA.

Section 2.6 describes the TP-OA setting in detail.

Table 12: Originator element

Element	**`<Originator>`**
Parent Element	`<MessageMT>`
Child Elements	–
Has Attributes	-
Character Data	*A string of numeric or alphanumeric characters with the restrictions defined in Section 2.6. eg. MAYFLOWER or +447740101097 (a number which can be either a handset or a dedicated MO number purchased through Dynmark)* *OR* *The string 'etxt' – the DMP will replace 'etxt' with an MO telephone number so that the recipient may reply to the message. The reply can be retrieved using the method described on section 4.* *The limit is 15 chars.*
Usage	Required
Occurrences	1

2.3.3.7 Content Element

The character data of the required `<Content>` element specifies the text or binary content of the SMS messages described by the message template.

Section 1.1 describes the restrictions on content for each SMS message type.

Table 13: Content element

Element	**`<Content>`**
Parent Element	`<MessageMT>`
Child Elements	–
Has Attributes	-
Character Data	*A string of characters with a minimum length of 1 character and a maximum length of 160 characters.* You can send personalized messages by using mailmerge. {MFn} identifies your mailmerge field where n is number from 1 to 9.
Usage	Required
Occurrences	1

2.3.3.8 MergeFieldOrder

The order in which merge field data will occur. Can be deleted if you do not have MergeField data.

Table 14: MergeFieldOrder element

Element	**`<MergeFieldOrder>`**
Parent Element	`<MessageMT>`
Child Elements	–
Has Attributes	-
Character Data	Eg: value="{MF1}~{MF2}~{MF3}~{MF4}"

Usage	optional
Occurrences	1

2.3.3.9 Recipient Element

The required `<Recipient>` element specifies a recipient of the SMS message described by the message template.

The optional `userRef` attribute allows the customer to associate a custom reference string with each SMS message generated by the message template. That is, each SMS message sent to a handset may be associated with a customer-specified reference string. This reference string is returned in the `<SmsMTSegmentResponse>` element of the DMP response, which is described in detail in Section 2.4.4.6.

The optional `mfv` element contains the mergefield data. Each element is separated by ~ delimiters and each block of data is separated by the #!# delimiter. The mergefield data has to be appear in the order given by MergeFieldOrder tag (see above).

The `<Number>` child element, described in the next section, allows specification of the handset number.

Table 15: Recipient element

Element	**`<Recipient>`**
Parent Element	`<MessageMT>`
Child Elements	`<Number>`
Has Attributes	`userRef`
Character Data	-
Usage	Required
Occurrences	1 to unlimited
Attribute	**`userRef`**
Type	string
Usage	optional
Values	*Any character permitted by the XML Schema string type may be used; the length of the string value must be between 0 and 40 characters.*
Default Value	–
Attribute	**`o`**
Type	string
Usage	required
Values	*“” – not used, but insert blank string*
Default Value	–
Attribute	**`mfv`**
Type	string
Usage	optional
Values	*Eg:* ="Simon~Fernando~CJ2GH~fieldOJ#!#John~Hamilton~UB60PU~fieldJH#!#firstname~lastname~zipcode~newfield#!#"
Default Value	–

2.3.3.10 Number Element

The character data of the required `<Number>` element specifies the number of the destination handset, in fully qualified international dialling format, represented by the parent `<Recipient>` element. See the table below for restrictions on the character data of this element.

For the hypothetical local UK handset number `01234567890,` the required character data is `441234567890.` That is, the leading zero is stripped from the local handset number, and the country code `44` is prefixed to the number to form the fully qualified international dialling number.

It is the responsibility of the client application to ensure that supplied handset numbers are valid prior to submission.

Numbers for each recipient are separated by the comma (,) delimiter. Numbers must be in the same order as the mergfield data to ensure the correct merged MT Message is sent to the intended handset. Not the final comma at the end of the list.

Table 16: Number element

Element	**`<Number>`**
Parent Element	`<Recipient>`
Child Elements	-
Has Attributes	-
Character Data	*A string of numeric characters (digits 0 to 9 only) specifying the number of the destination handset represented by the parent* `<Recipient>` *element. Note that the number must be specified in fully qualified international dialling format, including country code and area code. The leading '+' character, used by handsets to prefix a country code, is NOT permitted.* *The minimum length of the number is 7 characters, and the maximum length of the number is 15 characters.* *Eg:* 4472341234,4482341234,447971893841, (note the final comma)
Usage	Required
Occurrences	1

2.4 Response

2.4.1 Server HTTP Response

Following successful submission of a message sending request to the DMP, the server returns the HTTP status code 200 ('OK') as an indication that the submission was received and processed.

If the client application receives any HTTP status code other than 200, then it is the responsibility of the client application to take appropriate action to deal with the condition indicated by the code. If a Dynmark error is indicated, please send an e-mail describing the error to support@dynmark.com

Please refer to reference [HTTP] for the definition of the HTTP status codes.

2.4.2 HTTP Interface Response

Following successful submission of a message sending request to the DMP, the HTTP interface returns a synchronous response to the client application as an XML document. The remainder of this section describes the format of this document.

It should be noted that the content of the response document varies depending on whether the whole submission was accepted and successfully processed by the DMP, or errors were encountered during processing. In the event of ANY error occurring during processing, the ENTIRE submission is rejected with an appropriate return code indicating the nature of the error.

2.4.3 Example Response

Listing 2 shows an example of a successful HTTP interface response. The XML constitutes the body of the HTTP response. Note that some attributes have been further indented to improve the clarity of the listing.

Listing 2: XML document showing example of message sending response

```
<?xml version="1.0" encoding="utf-8"?>
<Dmp version="1.0">
 <MessageMTSendResponse returnCode="Success" returnDescription="All messages
accepted.">
      <MessageMTResponse>
          <SmsMTResponse>
             <SmsMTSegmentResponse userRef="USERNAME"
                  dynmarkRef="42993D4CF9A448BBA845D3ED0E1C0CDC" />
          </SmsMTResponse>
      </MessageMTResponse>
   </MessageMTSendResponse>
  </Dmp>

else for example:

<?xml version="1.0" encoding="utf-8"?>
<Dmp version="1.0">
<MessageMTSendResponse returnCode="ErrUserPasswordInvalid"
returnDescription="The supplied password is invalid." returnDetail="knp30x798" />
   </Dmp>
```

The XML format shown in Listing 2 is described in detail in the following section. Note that the values for the `dynmarkRef` attributes are for illustrative purposes only and will vary in a real application.

2.4.4 XML Document Format

2.4.4.1 XML Declaration

The `<?xml>` element is required to identify the document type to the client application and it forms the first line of each server response.

The value of the required `version` attribute is always `'1.0'`, and the value of the required `encoding` attribute is always `'utf-8'`.

Table 17: XML declaration

Element	**`<?xml>`**
Parent Element	-
Has Attributes	`version` `encoding`
Child Elements	-
Character Data	-
Attribute	**`version`**
Parent Element	`<?xml>`
Type	string
Usage	required
Values	must be `"1.0"`
Default Value	-
Occurrences	1
Attribute	**`encoding`**
Parent Element	`<?xml>`
Type	string
Usage	required
Values	must be `"utf-8"`
Default Value	-
Occurrences	1

2.4.4.2 Dmp Element

The required `<Dmp>` element is the document root element and represents a generic response from the DMP.

The required `version` attribute is used to ensure compatibility between different versions of the DMP. Its current value is `'1.0'`.

Table 18: Dmp element

Element	**`<Dmp>`**
Parent Element	-
Child Elements	`<MessageMTSendResponse>`
Has Attributes	`version`
Character Data	-
Usage	Required
Occurrences	1
Attribute	**`encoding`**
Type	string
Usage	required
Values	must be `'1.0'`
Default Value	-

2.4.4.3 MessageMTSendResponse Element

The required `<MessageMTSendResponse>` element represents the DMP's response to the message sending request.

The required `returnCode` attribute defines the DMP response as a code string that can be processed by the client application.

The required `returnDescription` attribute is intended as a human-readable string that describes the meaning of the `returnCode` attribute.

These attributes are described in detail in Section 2.4.5.

Table 19: MessageMTSendResponse element

Element	**`<MessageMTSendResponse>`**
Parent Element	`<Dmp>`
Child Elements	`<MessageMTResponse>`

Has Attributes	`returnCode`
	`returnDescription`
	`returnDetail`

Character Data	-
Usage	Required
Occurrences	1
Attribute	**`returnCode`**
Type	string
Usage	required
Values	*See Section 2.4.5.2 for permitted values*
Default Value	–
Attribute	**`returnDescription`**
Type	string
Usage	required
Values	*See section 2.4.5.3 for permitted values*
Default Value	–

2.4.4.4 MessageMTResponse Element

The optional `<MessageMTResponse>` element represents the DMP's response to the message template (represented by the `<MessageMT>` element) submitted in the request.

If processing of the message submission was unsuccessful, then this element and its child elements are not present in the response.

Table 20: MessageMTResponse element

Element	**`<MessageMTResponse>`**
Parent Element	`<MessageMTSendResponse>`
Child Elements	`<SmsMTResponse>`
Has Attributes	-
Character Data	-
Usage	Optional
Occurrences	1 in the event of successful processing of the message submission, 0 in the event of unsuccessful processing of the message submission

2.4.4.5 SmsMTResponse Element

The required `<SmsMTResponse>` element represents a generated SMS message.

Table 21: SmsMTResponse element

Element	**`<SmsMTResponse>`**
Parent Element	`<MessageMTResponse>`
Child Elements	`<SmsMTSegmentResponse>`
Has Attributes	-
Character Data	-
Usage	Required
Occurrences	1

2.4.4.6 SmsMTSegmentResponse Element

At present, each `<SmsMTResponse>` element (described in Section 2.4.4.5) contains exactly one `<SmsMTSegmentResponse>` element. Further use of this element is reserved for future expansion of the response.

The required `dynmarkRef` attribute provides the customer with a unique reference for each SMS message generated by the request. The attribute's value is a 32-character hexadecimal-encoded string that is guaranteed to be unique. The client application should record the value of this attribute for each generated SMS message, as it is the only unique identifier for SMS messages generated by the DMP.

When contacting Dynmark support with queries relating to specific SMS messages generated by the DMP, customers should always quote the `dynmarkRef` values returned by the DMP for the SMS messages concerned.

Table 22: SmsMTSegmentResponse element

Element	**`<SmsMTSegmentResponse>`**
Parent Element	`<SmsMTResponse>`
Child Elements	-
Has Attributes	`userRef` `dynmarkRef`
Character Data	-
Usage	Required
Occurrences	1

Attribute	**`userRef`**
Type	string
Usage	required
Values	*Any character permitted by the XML Schema string type may be used; the length of the string value must be between 0 and 40 characters.*
Default Value	–
Attribute	**`dynmarkRef`**
Type	string
Usage	required
Values	*A string containing exactly 32 characters restricted to the digits 0 to 9 and the upper case letters A to F*
Default Value	–

2.4.5 Response Return Codes

2.4.5.1 Overview

The `<MessageMTSendResponse>` element of the HTTP interface response always contains the `returnCode` and `returnDescription` attributes, and may contain an optional `returnDetail` attribute, as described in Section 2.4.4.3.

This section describes in detail the possible values of these attributes.

2.4.5.2 Return Code Values

The required `returnCode` attribute provides a machine-readable code string indicating the status of the message submission response.

Table 23 lists the possible values of the `returnCode` attribute together with a description of the meaning of each attribute value.

It is important to note that ANY error that occurs during processing of the message submission request results in immediate termination of processing and rejection of the ENTIRE submission. In this case, the response will contain no `<MessageMTResponse>` elements. Only the FIRST error encountered is indicated by the return code value. It is possible that the submission may contain further errors that will not be indicated until the first error is fixed and the request is resubmitted.

In the event that the ENTIRE message submission is successfully processed, the return code will have the value `Success`, and the response will contain at least one `<MessageMTResponse>` element.

Table 23: Return code values and descriptions

Return Code Value	Description
`Success`	All messages accepted.
`ErrInternalRetryable`	An internal error occurred. Please resubmit the XML document.
`ErrInternalFatal`	A fatal internal error occurred. Please notify Dynmark support.
`ErrXmlBadlyFormed`	The submitted document contains badly-formed XML.
`ErrXmlInvalid`	The submitted XML document is not valid against the XML Schema.
`ErrAccountNonExistent`	The specified account does not exist.
`ErrAccountPasswordInvalid`	The supplied password is invalid.
`ErrAccountIPAddressNotAuthorised`	The source IP address is not authorised for this account.
`ErrOriginatorInvalid`	The originator value is invalid.
`ErrContentZeroLength`	The message content length is zero.
`ErrContentMaxLength`	The maximum message content length for the message type was exceeded.
`ErrContentInvalid`	The message content is invalid for the message type.
`ErrUserRefMaxLength`	The maximum user reference length was exceeded.
`ErrNumberLength`	The mobile number length is incorrect.
`ErrNumberFormat`	The mobile number format is incorrect.
`ErrNumberBlockedByAccount`	The mobile number is blocked for this account.
`ErrNumberBlockedByDynmark`	The mobile number is blocked by Dynmark.
`ErrTypeInvalid`	The message type value is invalid.
`ErrClassInvalid`	The message class value is invalid.

ErrAccountDailySmsLimitExceeded	The daily SMS limit for this account would have been exceeded.
ErrAccountMonthlySmsLimitExceeded	The monthly SMS limit for this account would have been exceeded.
ErrAccountCreditsInsufficient	The account has insufficient remaining credits.

2.4.5.3 Return Description Values

The required returnDescription attribute is intended as a human-readable string that describes the meaning of the returnCode attribute. The Description column of Table 23 gives the value of the returnDescription attribute for each returnCode value.

2.5 *Message Types*

2.5.1 Overview

Each message template in the request specifies a message type using the optional `type` attribute, as described in Section 2.3.3.5. The value of this attribute dictates how the DMP interprets the content of each message, and determines the type of SMS message sent to the handsets specified in the message template.

This section describes the range of message types currently supported by the HTTP interface <currently just 1, we no longer support ringtones or images>.

For each message type, a table shows the maximum character data length of the message template's `<Content>` element, together with the range of characters that the data may contain.

2.5.2 Text.GSM0338

2.5.2.1 Type Description

The `Text.GSM0338` message type is the default message type used when the `type` attribute is omitted from the message template element.

This type specifies a 7-bit text message containing a subset of the characters contained in the GSM default alphabet [GSM0338].

Table 24: Text.GSM0338 message type

Message Type	**`Text.GSM0338`**
Description	7-bit text message containing a subset of the characters contained in the GSM 0338 default alphabet [GSM0338]
Maximum Content Length	160
Content Character Data	*see* Table 25

2.5.2.2 Restricted GSM Default Alphabet

Table 25 shows the full set of characters that may be sent in the content of a `Text.GSM0338` message, together with the UTF-8 hexadecimal character code for each character (see reference [UNICODE]).

Table 25: Restricted GSM default alphabet

@		SP	0	¡	P	¿	p
40		20	30	A1	50	BF	70
£	_	!	1	A	Q	a	q
A3	5F	21	31	41	51	61	71
$		"	2	B	R	b	r
24		22	32	42	52	62	72
¥		#	3	C	S	c	s
A5		23	33	43	53	63	73
è		¤	4	D	T	d	t
E8		A4	34	44	54	64	74
é		%	5	E	U	e	u
E9		25	35	45	55	65	75
ù		&	6	F	V	f	v
F9		26	36	46	56	66	76
ì		'	7	G	W	g	w
EC		27	37	47	57	67	77
ò		(	8	H	X	h	x
F2		28	38	48	58	68	78
Ç		)	9	I	Y	i	y
C7		29	39	49	59	69	79
LF		*	:	J	Z	j	z
0A		2A	3A	4A	5A	6A	7A
Ø		+	;	K	Ä	k	ä
D8		2B	3B	4B	C4	6B	E4
ø	Æ	,	<	L	Ö	l	ö
F8	C6	2C	3C	4C	D6	6C	F6
CR	æ	-	=	M	Ñ	m	ñ
0D	E6	2D	3D	4D	D1	6D	F1
Å	ß	.	>	N	Ü	n	ü
C5	DF	2E	3E	4E	DC	6E	FC
å	É	/	?	O	§	o	à
E5	C9	2F	3F	4F	A7	6F	E0

Special attention is drawn to the following characters in Table 25:

- Character code C7 represents the upper case character Ç, not the lower case character ç, as most GSM handsets do not support the latter.
- There are ten upper case Greek characters present in the GSM default alphabet that are not currently supported by the DMP. These are the Greek upper case characters which cannot be represented by capital Latin characters.
- The GSM default alphabet escape character, ESC, is not currently supported by the DMP, with the result that the DMP does not currently support the GSM default alphabet extension characters.

2.5.2.3 Encoding

The GSM default alphabet characters supported by the DMP are represented in the character data of the `<Content>` element using UTF-8 encoding.

Most of the supported characters can be represented as character literals. The exceptions to this are the five special markup characters described in Section 2.5.2.4, and the line feed (LF) and carriage return (CR) characters described in Section 2.5.2.5.

All characters may be represented using the appropriate character reference formed from the UTF-8 hexadecimal character code given in Table 25.

For example, the character É may be represented using the character reference &#xC9;

2.5.2.4 Special Markup Characters

Five characters have special meaning in XML markup, and each must be represented in the character data of the `<Content>` element using the appropriate entity reference or character reference for that character as shown in Table 26.

Table 26: Special markup characters

Character	Description	Entity Reference	Character Reference
`<`	Less-than sign	`&lt;`	`&#x3C;`
`>`	Greater-than sign	`&gt;`	`&#x3E;`
`&`	Ampersand	`&amp;`	`&#x26;`
`'`	Apostrophe	`&apos;`	`&#x27;`
`"`	Quotation mark	`&quot;`	`&#x22;`

2.5.2.5 Line Breaks

Handsets vary in their ability to display line breaks in SMS message content. However, most GSM handsets will display a line break in response to the line feed (LF) character, which must be represented in the character data of the `<Content>` element using any one of the following sequences of character references:

- `&#x0A;`
- `&#x0D;`
- `&#x0D;&#x0A;`

Whilst [GSM0338] also supports the carriage return (CR) character, use of this character is not recommended as it is not well supported by modern handsets.

2.5.2.6 Encoding Example

Listing 3 shows an example of a `<Content>` element used to send an SMS message of type `Text.GSM0338` which forces a line break and displays the character É using a character reference.

Listing 3: Text.GSM0338 encoding example

```
<Content>Here is a line break:&#x0A;Here is a capital E with acute: &#xC9;</Content>
```

Listing 4 shows the resulting text message as displayed on most handsets. Note that line wrapping may vary.

Listing 4: Text.GSM0338 encoding example handset text

```
Here is a line break:
Here is a capital E with acute: É
```

2.6 TP-OA Setting

As described in Section 2.3.3.6, each message template in the request requires a single `<Originator>` element to specify the TP-Originating-Address (TP-OA) of the SMS messages described by the template.

[GSM0340] defines TP-OA as the "address of the originating SME [Short Messaging Entity]". Most handsets display the TP-OA value as the 'from' field of an SMS message. The TP-OA of an SMS message sent through the DMP can be set to either a numeric value or an alphanumeric value using the `<Originator>` element.

Table 27 describes the restrictions on the character data of the `<Originator>` element.

Table 27: Originator character data restrictions

Originator Type	Min String Length	Max String Length	Permitted Characters
Numeric	2	15	0 to 9
Alphanumeric	2	11	0 to 9 a to z A to Z *(hyphen)* - *(underscore)* _ *(period)* . *(space)*

It is important to note that support for originator characters may vary with downstream provider and operator. Customers should therefore send test messages to selected handsets to ensure that the desired characters can be transmitted.

3 Message Receiving Interface Specification

3.1 Overview

MO messages received by the DMP are delivered to customers by means of an XML document transmitted in response to an HTTP POST request to an endpoint URL. This section section provides all the details necessary to use this interface from a client application.

3.2 Customer Gateway Accounts

3.2.1 Overview

Each customer wishing to receive MO messages via the DMP must have at least one authorised DMP message account.

Customers wishing to purchase their own dedicated MO number should contact sales@dynmark.com and must agree with Dynmark the gateway account configuration parameters described below.

3.2.2 Authentication Details

The customer must agree with Dynmark a username and password that will be transmitted in each DMP MO message notification transmitted to the customer in an HTTP request. This username and password is the same as those used for MT (above). The customer can use these details to carry out simple authentication on the validity of the request.

3.2.3 Endpoint

MO messages received by the DMP are delivered to customers by means of an XML document transmitted in response to an HTTP POST request to an endpoint URL. The endpoint URL is given in table Table 28: URL of EndPoint for receiving MO messages.

Table 28: URL of EndPoint for receiving MO messages

http://80.82.139.132/services/ServiceEndPoint.aspx

3.3 Request

3.3.1 Request Transmission

The client application initiates a request to send single MT messages by opening a connection to the HTTP interface server using the settings listed in Table 29: Server settings for message receiving.

Table 29: Server settings for message receiving

Server Name	http://80.82.139.132
Server Port	80
Protocol	TCP

The client application must then transmit an HTTP POST request to the server using the HTTP settings listed in Table 30.

Table 30: HTTP POST settings for message receiving

HTTP POST Request URI	/services/ServiceEndPoint.aspx
HTTP Content Type	text/xml
HTTP Content Length	*<length of HTTP request body>*

The HTTP POST request body will contain an XML document which conforms to the specification given in Section 3.3.2.1.

3.3.2 Example Request To Receive MO Messages

Listing 5 shows an example of an XML request used to receive an MO message.

The XML shown constitutes the body of the HTTP POST request.

Listing 5: XML document showing example of message receiving request

```
<methodCall>
<methodName>GetMOMessages</methodName>
<params>
<param type='String'>etxtusername</param>
<param type='String'>password</param>
<param type='Int32'>146655</param>
</params>
</methodCall>
```

The XML format shown in Listing 5 is described in detail in the following section.

Note that in this listing the values for account name and account password are for illustrative purposes only and would need to be replaced by actual values in a real application.

3.3.2.1 XML Document Format

3.3.2.1.1 XML Declaration

The `<?xml>` element is required to identify the document type to the server, and appears as the first line in each submission. No other characters precede this element.

The required `version` attribute has the value `'1.0'`, and the required `encoding` attribute has the value `'utf-8'`.

Table 31: XML declaration

Element	**`<?xml>`**
Parent Element	-
Has Attributes	`version` `encoding`
Child Elements	methodCall
Character Data	-
Attribute	**`version`**
Parent Element	`<?xml>`
Type	string
Usage	required
Values	must be "`1.0`"
Default Value	-
Occurrences	1
Attribute	**`encoding`**
Parent Element	`<?xml>`
Type	string
Usage	required
Values	must be "`utf-8`"
Default Value	-
Occurrences	1

3.3.2.2 MethodCall Element

The required `<MethodCall>` element is the root element for extended DMP methods and attributes.

Table 32: MethodCall element

Element	**`<MethodCall>`**
Parent Element	-

Child Elements	`<methodName>`
	`<params>`

Has Attributes	`none`
Character Data	-
Usage	Required
Occurrences	1

3.3.2.3 MethodName Element

The required `<MethodName>` element represents the name of the required extended DMP method.

Table 33: MethodName element

Element	**`<MethodName>`**
Parent Element	`<MethodCall>`
Child Elements	-
Has Attributes	`none`
Character Data	Name of method. Eg: 'GetMOMessages' to retrieve MO messages, or 'UpdateMOMessageStatus' to mark messages as downloaded or deleted.
Usage	required
Occurrences	1

3.3.2.4 Params Element

The required `<Params>` element represents the parameters passed to the extended DMP method.

Table 34: Params element

Element	**`<Params>`**
Parent Element	`<MethodCall>`
Child Elements	–
Has Attributes	-
Character Data	There are 3 Parameters for the GetMOMessages method Username: <param type='string'>etxtusername</param> Password: <param type='string'>etxtpassword</param> Last downloaded MO: <param type='Int32'>123456</param>
Usage	All Params Required
Occurrences	1

3.3.3 Example Request To Update MO Status

Listing 7 shows an example of an XML request used to update the status of MO messages on the DMP. Messages can be marked as downloaded or deleted.

The XML shown constitutes the body of the HTTP POST request.

Listing 6: XML document showing example of message receiving request

```
<methodCall>
<methodName>UpdateMOMessageStatus</methodName>
<params>
<param type='String'>etxtusername</param>
<param type='String'>password</param>
<param type='Int32'>146655</param>
<param type='Int32'>146657</param>
<param type='Boolean'>true</param>
</params>
</methodCall>
```

The XML format shown in Listing 5 is described in detail in the following section.

Note that in this listing the values for account name and account password are for illustrative purposes only and would need to be replaced by actual values in a real application.

3.3.3.1 XML Document Format

3.3.3.1.1 XML Declaration

The `<?xml>` element is required to identify the document type to the server, and appears as the first line in each submission. No other characters precede this element.

The required `version` attribute has the value `'1.0'`, and the required `encoding` attribute has the value `'utf-8'`.

Table 35: XML declaration

Element	**`<?xml>`**
Parent Element	-
Has Attributes	`version` `encoding`
Child Elements	methodCall
Character Data	-
Attribute	**`version`**
Parent Element	`<?xml>`
Type	string
Usage	required
Values	must be "`1.0`"
Default Value	-
Occurrences	1
Attribute	**`encoding`**
Parent Element	`<?xml>`
Type	string
Usage	required
Values	must be "`utf-8`"
Default Value	-
Occurrences	1

3.3.3.2 MethodCall Element

The required `<MethodCall>` element is the root element for extended DMP methods and attributes.

Table 36: MethodCall element

Element	**`<MethodCall>`**
Parent Element	-

Child Elements	`<methodName>`
	`<params>`

Has Attributes	`none`
Character Data	-
Usage	Required
Occurrences	1

3.3.3.3 MethodName Element

The required `<MethodName>` element represents the name of the required extended DMP method.

Table 37: MethodName element

Element	**`<MethodName>`**
Parent Element	`<MethodCall>`
Child Elements	-
Has Attributes	`none`
Character Data	Name of method. Eg: 'GetMOMessages' to retrieve MO messages, or 'UpdateMOMessageStatus' to mark messages as downloaded or deleted.
Usage	required
Occurrences	1

3.3.3.4 Params Element

The required `<Params>` element represents the parameters passed to the extended DMP method.

Table 38: Params element

Element	**`<Params>`**
Parent Element	`<MethodCall>`
Child Elements	–
Has Attributes	-
Character Data	There are 5 Parameters for the UpdateMOMessageStatus method Username: <param type='string'>etxtusername</param> Password: <param type='string'>etxtpassword</param> Start MO ID: <param type='Int32'>123456</param> End MO ID: <param type='Int32'>123459</param> End MO ID: <param type='Boolean'>true</param> When this final parameter is=true, the mo message is marked as DELETED. Therefore in future you will not be able to download this MO message again. If this parameter is 'false' then these messages at our server are marked only as DOWNLOADED. In this scenario, if you request for MOmessage and send ID=0, then the MO message will be downloaded again.

Usage	All Params Required
Occurrences	1

Note that messages shoud be marked deleted by UpdateMOMessageStatus or you should maintain MaxMessageId downloaded and send it in GetMOMessages to avoid getting same MO messages again.

3.4 Response

3.4.1 Server HTTP Response

Following successful submission of a message receiving request to the DMP, the server returns the HTTP status code 200 ('OK') as an indication that the submission was received and processed.

If the client application receives any HTTP status code other than 200, then it is the responsibility of the client application to take appropriate action to deal with the condition indicated by the code. If a Dynmark error is indicated, please send an e-mail describing the error to support@dynmark.com

Please refer to reference [HTTP] for the definition of the HTTP status codes.

3.4.2 HTTP Interface Response

Following successful submission of a message sending request to the DMP, the HTTP interface returns a synchronous response to the client application as an XML document. The remainder of this section describes the format of this document.

It should be noted that the content of the response document varies depending on whether the whole submission was accepted and successfully processed by the DMP, or errors were encountered during processing. In the event of ANY error occurring during processing, the ENTIRE submission is rejected with an appropriate return code indicating the nature of the error.

3.4.3 Example Response

Listing 8 shows an example of a successful HTTP interface response. The XML constitutes the body of the HTTP response. Note that some attributes have been further indented to improve the clarity of the listing.

Listing 7: XML document showing example of message receiving response (this is XML representation of ADO.net dataset):

```
- <methodResponse>
- <result>
- <value type="complex">
- <UserMOMessage xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <status>Success</status>
- <ds>
- <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
     xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
- <xs:element name="NewDataSet" msdata:IsDataSet="true">
- <xs:complexType>
- <xs:choice maxOccurs="unbounded">
- <xs:element name="MOMessage">
- <xs:complexType>
- <xs:sequence>
  <xs:element name="ID" type="xs:int" minOccurs="0" />
  <xs:element name="F_ACCOUNT_ID" type="xs:int" minOccurs="0" />
  <xs:element name="F_SMS_DIRECTION_ID" type="xs:int" minOccurs="0" />
  <xs:element name="PE_SMS_ID" type="xs:int" minOccurs="0" />
  <xs:element name="F_MESSAGE_ID" type="xs:int" minOccurs="0" />
  <xs:element name="F_CARRIER_ID" type="xs:int" minOccurs="0" />
  <xs:element name="F_GATEWAY_ACCOUNT_ID" type="xs:int" minOccurs="0" />
  <xs:element name="F_SMS_TYPE_ID" type="xs:int" minOccurs="0" />
  <xs:element name="F_SMS_RESPONSE_ID" type="xs:int" minOccurs="0" />
  <xs:element name="F_SMS_STATUS_ID" type="xs:int" minOccurs="0" />
  <xs:element name="F_SMS_DISPOSITION_ID" type="xs:int" minOccurs="0" />
  <xs:element name="F_SMS_TRACKING_ID" type="xs:int" minOccurs="0" />
  <xs:element name="F_FUNNEL_ID" type="xs:int" minOccurs="0" />
  <xs:element name="F_SMS_RETRY_PLAN_ID" type="xs:int" minOccurs="0" />
  <xs:element name="CLIENT_REFERENCE" type="xs:string" minOccurs="0" />
  <xs:element name="SERVER_UNIQUE_REFERENCE" type="xs:string" minOccurs="0" />
  <xs:element name="GATEWAY_UNIQUE_REFERENCE" type="xs:string" minOccurs="0" />
  <xs:element name="MSISDN" type="xs:string" minOccurs="0" />
  <xs:element name="ORIGINATOR" type="xs:string" minOccurs="0" />
  <xs:element name="PAYLOAD" type="xs:string" minOccurs="0" />
  <xs:element name="SMS_CLASS" type="xs:unsignedByte" minOccurs="0" />
  <xs:element name="SEGMENT_NUMBER" type="xs:unsignedByte" minOccurs="0" />
  <xs:element name="DELIVER_AFTER" type="xs:dateTime" minOccurs="0" />
  <xs:element name="CREATED" type="xs:dateTime" minOccurs="0" />
  <xs:element name="MODIFIED" type="xs:dateTime" minOccurs="0" />
  <xs:element name="DELETED" type="xs:boolean" minOccurs="0" />
  <xs:element name="F_DOWNLOAD_STATUS_ID" type="xs:int" minOccurs="0" />
  <xs:element name="F_NOTIFICATION_STATUS_ID" type="xs:int" minOccurs="0" />
  <xs:element name="TODEL_TRANSACTION_ID" type="xs:string" minOccurs="0" />
  <xs:element name="TRANSACTION_ID" type="xs:int" minOccurs="0" />
  <xs:element name="STATUS_UPDATE_TIME" type="xs:dateTime" minOccurs="0" />
  <xs:element name="f_rb_carrier_id" type="xs:int" minOccurs="0" />
  <xs:element name="f_rb_mo_account_id" type="xs:int" minOccurs="0" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
    </xs:choice>
    </xs:complexType>
    </xs:element>
    </xs:schema>
- <diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"
     xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
- <NewDataSet>
- <MOMessage diffgr:id="MOMessage1" msdata:rowOrder="0">
  <ID>146804</ID>
  <F_ACCOUNT_ID>16</F_ACCOUNT_ID>
  <F_SMS_DIRECTION_ID>2</F_SMS_DIRECTION_ID>
  <F_MESSAGE_ID>917719</F_MESSAGE_ID>
  <F_CARRIER_ID>3</F_CARRIER_ID>
  <F_GATEWAY_ACCOUNT_ID>11</F_GATEWAY_ACCOUNT_ID>
  <F_SMS_TYPE_ID>1</F_SMS_TYPE_ID>
  <F_SMS_RESPONSE_ID>85</F_SMS_RESPONSE_ID>
  <F_SMS_STATUS_ID>8</F_SMS_STATUS_ID>
  <F_SMS_DISPOSITION_ID>6</F_SMS_DISPOSITION_ID>
  <F_SMS_TRACKING_ID>1</F_SMS_TRACKING_ID>
  <F_FUNNEL_ID>2</F_FUNNEL_ID>
  <CLIENT_REFERENCE>sanshalim</CLIENT_REFERENCE>
  <SERVER_UNIQUE_REFERENCE>8D1221C714AE42BCBC91A25AA2354B66</SERVER_UNIQUE_REFERENCE>
  <MSISDN>919819433463</MSISDN>
  <ORIGINATOR>447786202506</ORIGINATOR>
  <PAYLOAD>Nokia test 5</PAYLOAD>
  <SMS_CLASS>0</SMS_CLASS>
  <SEGMENT_NUMBER>1</SEGMENT_NUMBER>
  <DELIVER_AFTER>2005-05-05T19:19:02.0430000+05:30</DELIVER_AFTER>
  <CREATED>2005-05-05T19:19:02.0430000+05:30</CREATED>
  <MODIFIED>2005-05-05T19:19:10.7500000+05:30</MODIFIED>
  <DELETED>false</DELETED>
  <F_NOTIFICATION_STATUS_ID>25</F_NOTIFICATION_STATUS_ID>
  <f_rb_carrier_id>0</f_rb_carrier_id>
  <f_rb_mo_account_id>0</f_rb_mo_account_id>
    </MOMessage>
    </NewDataSet>
    </diffgr:diffgram>
    </ds>
    </UserMOMessage>
    </value>
    </result>
    </methodResponse>
```

The XML format shown in Listing 8 is described in detail in the following section. Note that the values for all attributes are for illustrative purposes only and will vary in a real application.

3.4.4 XML Document Format

The following is a list of the tags contained within the DataSet with descriptions. Most of these tags are of little interest to customers connecting directly to the DMP. The tags which are relevant are highlighted.

Element	**`<ID>`**
Relevant	No
Parent Element	<ds> ADO.NET Dataset
Has Attributes	`None`
Child Elements	-
Character Data	Integer. Unique ID of MO Message

Element	**`<F_ACCOUNT_ID>`**
Relevant	No
Parent Element	<ds> ADO.NET Dataset
Has Attributes	`None`
Child Elements	-
Character Data	Integer. Unique ID of MO Message

Element	**`<F_SMS_DIRECTION_ID>`**
Relevant	No
Parent Element	<ds> ADO.NET Dataset
Has Attributes	`None`
Child Elements	-
Character Data	Integer. Unique ID of MO Message

Element	**`<F_MESSAGE_ID>`**
Relevant	***Yes***
Parent Element	<ds> ADO.NET Dataset
Has Attributes	`None`
Child Elements	-
Character Data	Integer. Unique ID of MO Message

Element	**`<F_CARRIER_ID>`**
Relevant	No
Parent Element	<ds> ADO.NET Dataset
Has Attributes	`None`
Child Elements	-
Character Data	Integer. Unique ID of MO Message

Element	**`<F_GATEWAY_ACCOUNT_ID>`**
Relevant	No
Parent Element	<ds> ADO.NET Dataset
Has Attributes	`None`
Child Elements	-
Character Data	Integer. Unique ID of MO Message

Element	**`<F_SMS_TYPE_ID>`**
Relevant	No
Parent Element	<ds> ADO.NET Dataset
Has Attributes	`None`
Child Elements	-
Character Data	Integer. Unique ID of MO Message

Element	**`<F_SMS_RESPONSE_ID>`**
Relevant	No
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	Integer. Unique ID of MO Message

Element	**`<F_SMS_STATUS_ID>`**
Relevant	No
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	Integer. Unique ID of MO Message

Element	**`<F_SMS_DISPOSITION_ID>`**
Relevant	No
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	Integer. Unique ID of MO Message

Element	**`<F_SMS_TRACKING_ID>`**
Relevant	No
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	Integer. Unique ID of MO Message

Element	**`<F_FUNNEL_ID>`**
Relevant	No
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	Integer. Unique ID of MO Message

Element	**`<CLIENT_REFERENCE>`**
Relevant	***Yes***
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	String. The account username.

Element	**`<SERVER_UNIQUE_REFERENCE>`**
Relevant	***Yes***
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	String. The unique reference on the DMP for an MO message.

Element	**`<MSISDN>`**
Relevant	***Yes***
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	String. The handset mobile number that sent an inbound message to the DMP.

Element	**<ORIGINATOR>**
Relevant	***Yes***
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	String. The MO number attached to the account. This will always be the same number for customers with their own dedicated inbound telephone number. This number may change for DMP allocated dynamic TP-OA.

Element	**<PAYLOAD>**
Relevant	***Yes***
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	String. The (max) 160 character text message.

Element	**<SMS_CLASS>**
Relevant	No
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	Integer. Class of SMS message. Always 0.

Element	**<SEGMENT_NUMBER>**
Relevant	***Yes***
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	Integer. The segment ID for messages that are split over multiple messages. This is for messages that are > 160 characters.

Element	**<DELIVER_AFTER>**
Relevant	No
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	DateTime. The date/time the message arrived on the platform.

Element	**<CREATED>**
Relevant	No.
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	DateTime. The date/time the message arrived on the platform.

Element	**<MODIFIED>**
Relevant	No.
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	DateTime. The date/time the message arrived on the platform.

Element	**<DELETED>**
Relevant	No.
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	Bool. Always false. If true, then customer would not be able to receive.

Element	**<F_NOTIFICATION_STATUS_ID>**
Relevant	No.
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	Integer. Used in MT for delivery status.

Element	**<f_rb_carrier_id>**
Relevant	No.
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	Integer. Internal premium code.

Element	**<f_rb_mo_account_id>**
Relevant	No.
Parent Element	<ds> ADO.NET Dataset
Has Attributes	None
Child Elements	-
Character Data	Integer. Internal premium code.

Appendix A: References

Reference	Title	URL
[HTTP]	*RFC 2616: Hypertext Transfer Protocol – HTTP/1.1*	ftp://ftp.isi.edu/in-notes/rfc2616.txt
[GSM0338]	*GSM 03.38 version 7.2.0 Release 1998*	http://www.etsi.org/
[GSM0340]	*GSM 03.40 version 7.4.0 Release 1998*	http://www.etsi.org/
[NOKSM]	*Nokia Smart Messaging Specification, Revision 3.0.0*	http://www.forum.nokia.com/main/1,35452,1_2_5_3,00.html
[NOKSMFAQ]	*Nokia Smart Messaging FAQ v2.0*	http://www.forum.nokia.com/main/1,35452,1_2_5_5,00.html
[XML]	*Extensible Markup Language (XML) 1.0 (Second Edition)*	http://www.w3.org/TR/REC-xml
[XMLNS]	*Namespaces in XML*	http://www.w3.org/TR/REC-xml-names/
[XMLS]	*XML Schema Part 0: Primer* *XML Schema Part 1: Structures* *XML Schema Part 2: Datatypes*	http://www.w3.org/XML/Schema
[UNICODE]	The Unicode specifications	http://www.unicode.org/