

MessageMedia

Messaging Web Service

April 2013

Table of contents

Section 1: Change History	5
Section 2: Glossary	5
Section 3: Introduction	6
3.1 MESSAGING WEB SERVICE FEATURES	6
3.2 CLIENT REQUIREMENTS FOR SERVICE INTEGRATION	6
Section 4: Messaging Web Service Overview.....	7
Section 5: XML Transport	7
5.1 XML TRANSPORT VIA SOAP	7
5.1.1 USING THE WSDL FILE TO GENERATING CLIENT CODE	8
5.1.1.1 GENERATING A CLIENT PROXY FOR .NET.....	8
5.1.1.2 GENERATING A CLIENT PROXY FOR JAVA.....	8
5.1.1.3 GSOAP FOR C/C++.....	8
5.1.2 SOAP REQUEST FORMAT	9
5.1.3 SOAP RESPONSE FORMAT	10
5.1.4 SOAP FAULT FORMAT	10
5.2 XML TRANSPORT VIA HTTP-POST	11
Section 6: XML Validation	12
Section 7: Messaging Requests and Responses.....	13
7.1 FAULT RESPONSE FORMAT	13
7.2 CHECK USER REQUEST.....	13
7.2.1 FORMAT OF THE CHECK USER REQUEST	13
7.2.2 EXPLANATION OF THE CHECK USER REQUEST FORMAT	14
7.3 CHECK USER RESPONSE	14
7.3.1 FORMAT OF THE CHECK USER RESPONSE	14
7.3.2 EXPLANATION OF THE CHECK USER RESPONSE FORMAT	14
7.4 SEND MESSAGES REQUEST	15
7.4.1 FORMAT OF THE SEND MESSAGES REQUEST	15
7.4.2 EXPLANATION OF THE SEND MESSAGES REQUEST	16
7.5 SEND MESSAGES RESPONSE	20
7.5.1 FORMAT OF THE SEND MESSAGES RESPONSE	20
7.5.2 EXPLANATION OF THE SEND MESSAGES RESPONSE	22
7.6 CHECK REPLIES REQUEST	23
7.6.1 FORMAT OF THE CHECK REPLIES REQUEST	23
7.6.2 EXPLANATION OF THE CHECK REPLIES REQUEST FORMAT	24

7.7 CHECK REPLIES RESPONSE	24
7.7.1 FORMAT OF THE CHECK REPLIES RESPONSE	24
7.7.2 EXPLANATION OF THE CHECK REPLIES RESPONSE FORMAT	25
7.8 CHECK REPORTS REQUEST	26
7.8.1 FORMAT OF THE CHECK REPORTS REQUEST	26
7.8.2 EXPLANATION OF THE CHECK REPORTS REQUEST FORMAT	26
7.9 CHECK REPORTS RESPONSE	27
7.9.1 FORMAT OF THE CHECK REPORTS RESPONSE	27
7.9.2 EXPLANATION OF THE CHECK REPORTS RESPONSE FORMAT	28
7.10 CONFIRM REPLIES REQUEST	29
7.10.1 FORMAT OF THE CONFIRM REPLIES REQUEST	29
7.10.2 EXPLANATION OF THE CONFIRM REPLIES REQUEST FORMAT	30
7.11 CONFIRM REPLIES RESPONSE	30
7.11.1 FORMAT OF THE CONFIRM REPLIES RESPONSE	31
7.11.2 EXPLANATION OF THE CONFIRM REPLIES RESPONSE FORMAT	31
7.12 CONFIRM REPORTS REQUEST	31
7.12.1 FORMAT OF THE CONFIRM REPORTS REQUEST	31
7.12.2 EXPLANATION OF THE CONFIRM REPORTS REQUEST FORMAT	32
7.13 CONFIRM REPORTS RESPONSE	32
7.13.1 FORMAT OF THE CONFIRM REPORTS RESPONSE	32
7.13.2 EXPLANATION OF THE CONFIRM REPLIES RESPONSE FORMAT	32
7.14 DELETE SCHEDULED MESSAGES REQUEST	33
7.14.1 FORMAT OF THE DELETE SCHEDULED MESSAGES REQUEST	33
7.14.2 EXPLANATION OF THE DELETE SCHEDULED MESSAGES REQUEST FORMAT	33
7.15 DELETE SCHEDULED MESSAGES RESPONSE	34
7.15.1 FORMAT OF THE DELETE SCHEDULED MESSAGES RESPONSE	34
7.15.2 EXPLANATION OF THE DELETE SCHEDULED MESSAGES RESPONSE FORMAT	34
7.16 BLOCK NUMBERS REQUEST	35
7.16.1 FORMAT OF THE BLOCK NUMBERS REQUEST	35

7.16.2 EXPLANATION OF THE BLOCK NUMBERS REQUEST FORMAT	35
7.17 BLOCK NUMBERS RESPONSE.....	36
7.17.1 FORMAT OF THE BLOCK NUMBERS RESPONSE	36
7.17.2 EXPLANATION OF THE BLOCK NUMBERS RESPONSE FORMAT	36
7.18 UNBLOCK NUMBERS REQUEST	37
7.18.1 FORMAT OF THE UNBLOCK NUMBERS REQUEST.....	37
7.18.2 EXPLANATION OF THE UNBLOCK NUMBERS REQUEST FORMAT	38
7.19 UNBLOCK NUMBERS RESPONSE	38
7.19.1 FORMAT OF THE UNBLOCK NUMBERS RESPONSE	38
7.19.2 EXPLANATION OF THE UNBLOCK NUMBERS RESPONSE FORMAT	39
7.20 GET BLOCKED NUMBERS REQUEST.....	39
7.20.1 FORMAT OF THE GET BLOCKED NUMBERS REQUEST.....	39
7.20.2 EXPLANATION OF THE GET BLOCKED NUMBERS REQUEST FORMAT	40
7.21 GET BLOCKED NUMBERS RESPONSE	40
7.21.1 FORMAT OF THE GET BLOCKED NUMBERS RESPONSE	40
7.21.2 EXPLANATION OF THE GET BLOCKED NUMBERS RESPONSE FORMAT	41
Appendix A: XML Schema Files.....	42
A.1 CHECK USER REQUEST SCHEMA	42
A.2 CHECK USER RESPONSE SCHEMA	42
A.3 SEND MESSAGES REQUEST SCHEMA	43
A.4 SEND MESSAGES RESPONSE SCHEMA.....	43
A.5 CHECK REPLIES REQUEST SCHEMA.....	43
A.6 CHECK REPLIES RESPONSE SCHEMA	44
A.7 CHECK REPORTS REQUEST SCHEMA.....	44
A.8 CHECK REPORTS RESPONSE SCHEMA	44
A.9 CONFIRM REPLIES REQUEST SCHEMA.....	45
A.10 CONFIRM REPLIES RESPONSE SCHEMA	45
A.11 CONFIRM REPORTS REQUEST SCHEMA	45
A.12 CONFIRM REPORTS RESPONSE SCHEMA.....	46
A.13 DELETE SCHEDULED MESSAGES REQUEST SCHEMA	46
A.14 DELETE SCHEDULED MESSAGES RESPONSE SCHEMA.....	46
A.15 BLOCK NUMBERS REQUEST SCHEMA	47

A.16 BLOCK NUMBERS RESPONSE SCHEMA	47
A.17 UNBLOCK NUMEBRS REQUEST SCHEMA	47
A.18 UNBLOCK NUMBERS RESPONSE SCHEMA.....	48
A.19 GET BLOCKED NUMBERS REQUEST SCHEMA	48
A.20 GET BLOCKED NUMBERS RESPONSE SCHEMA.....	48
A.21 FAULT RESPONSE SCHEMA	49
A.22 REQUEST TYPES SCHEMA.....	49
A.23 RESPONSE TYPES SCHEMA	53
A.24 COMMON TYPES SCHEMA	57
Appendix B: WSDL File	58
Appendix C: Redundant Service	65
Contact us.....	66

Section 1: Change History

Version	Date	XML Namespace	Change description
1.0	19 th November 2009	http://xml.m4u.com.au/2009	First release of the MessageMedia Messaging Web Service.
1.1	26 th September 2011	http://xml.m4u.com.au/2009	Additions for Delete Scheduled Message request & response.
1.2	16 th January 2012	http://xml.m4u.com.au/2009	Added blocked-numbers management operations.
1.3	5 th April 2012	http://xml.m4u.com.au/2009	Added source number masking support.
1.4	11 th July 2012	http://xml.m4u.com.au/2009	Added sendMode support.
1.5	5 th March 2013	http://xml.m4u.com.au/2009	Added tags support

Table 1.1: Change History

Section 2: Glossary

The table below provides a list of terms used within this document and their corresponding definitions.

Term	Definition
API	Application Programming Interface
<i>Concatenated SMS</i>	A method of overcoming the 160-character limit of SMS messages: messages greater than 160 characters in length are split up and then recombined at the receiving end to form a single long message
<i>Delivery Report</i>	A report that provides the SMS sender with the delivery status of a previously sent SMS message
<i>Document/Literal Wrapped Web Service</i>	A convention for structuring the web service description/interface (WSDL file and associated XML schema files) that yields the best results for interoperability with third party SOAP toolkits
<i>HTTP</i>	Hyper Text Transport Protocol
<i>HTTP-POST</i>	A method for transporting data to/from a web server via a HTTP request
<i>MessageMedia Gateway</i>	An internet server whose purpose is to route SMS and voice messages to communications carriers and receive SMS replies from those carriers
<i>MessageMedia Messaging Web Service</i>	A web service that provides an interface to the MessageMedia gateway
<i>MO: Mobile Originated</i>	Messages that are sent from a mobile device
<i>MT: Mobile Terminated</i>	Messages that are sent to a mobile device
<i>SMS</i>	Short Message Service
<i>SOAP</i>	A platform independent, language independent, standardised protocol for exchanging XML documents to and from a web service
<i>SOAP Client Proxy</i>	A software module that models a particular web service and allows the client application to communicate with that web service via function calls
<i>SOAP Toolkit</i>	One or more software applications that are used to parse a web service description (WSDL file and associated XML schema files) and automate the process of creating of creating a SOAP client proxy
<i>URL</i>	Uniform Resource Locator
<i>URL-encoded</i>	An encoding method in which certain special/restricted characters are replaced by an escape sequence
<i>Voice Message</i>	A message that is read out to the recipient by a text-to-

	speech software application
<i>Web Service</i>	A software system designed to support interoperable machine-to-machine interaction over a network
<i>WSDL</i>	Web Services Description Language
<i>WSDL File</i>	A XML file that describes the operations that a web service offers
<i>WS-I Basic Profile</i>	A specification from the Web Services Interoperability industry consortium (WS-I) that provides interoperability guidance for web service specifications
<i>XML</i>	Extensible Markup Language
<i>XML Namespace</i>	A construct for providing uniquely named elements and attributes in a XML document
<i>XML Schema</i>	A description of a XML document that expresses constraints on the structure and content of documents of that type

Table 2.1: Glossary of Terms

Section 3: Introduction

The Messaging Web Service provided by MessageMedia allows client applications to send and receive SMS and voice messages over the internet. This document provides the technical information necessary for clients to integrate this web service into their applications.

3.1 MESSAGING WEB SERVICE FEATURES

The Messaging Web Service provides client applications with the following features:

- Communication with the MessageMedia gateway via XML over either SOAP or HTTP-POST
- Communication over either secure or non-secure HTTP
- Support of mobile-terminated (MT) and mobile-originated (MO) messaging
- Ability to send voice messages to both mobile and landline devices
- Support of multiple mixed-format message sending per transaction
- Support of delivery reporting (delivery status information for MT messages returned by the carrier network)
- Ability to receive multiple MO messages per transaction
- Support of concatenated SMS (i.e. SMS messages greater than 160 characters are not split up—they are concatenated into a single SMS message delivered to the recipient)
- Redundant service(see Appendix C for details)

3.2 CLIENT REQUIREMENTS FOR SERVICE INTEGRATION

The Messaging Web Service provides a server-side interface to client applications. It can be leveraged by any programming language and platform that can transfer XML data over the internet via either SOAP or generic HTTP-POST. Following are the minimum level of requirements necessary for integration of the web service:

- General web programming knowledge and experience
- Basic familiarity with either SOAP or HTTP-POST
- Basic understanding of XML

-
- Computing platform and language that can communicate over the internet

Section 4: Messaging Web Service Overview

The Messaging Web Service provided by MessageMedia is a web service that allows client applications to send and receive SMS and voice messages over the internet. This web service provides an interface to the MessageMedia Gateway. The MessageMedia Gateway provides intelligent carrier-switching and routes messages through different communication carriers to maximise efficiency and service.

The Messaging Web Service is provided as a server-side API that may be integrated into client applications. It provides client applications with the ability to use the MessageMedia Gateway to send SMS and voice messages, receive SMS reply messages, and track delivery of SMS messages.

Client applications communicate with the Messaging Web Service via XML documents. These XML documents may be transported to and from the web service via either SOAP or HTTP-POST. These methods of XML transport are covered in Section 5.

Section 5: XML Transport

Client applications communicate with the MessageMedia Messaging Web Service via XML request and response documents. The XML documents that are exchanged between the client and the web service may be transported using either the SOAP protocol or the HTTP-POST protocol. Both methods of transportation are described in this section.

5.1 XML TRANSPORT VIA SOAP

The Messaging Web Service is available as a SOAP 1.1 web service at <http://soap.m4u.com.au>. In general, it is much easier for developers to integrate with SOAP web services as opposed to communication via XML over HTTP-POST. The reason for this is that there are an abundance of third-party toolkits for generating client proxies for XML web services.

A client proxy is a software module (usually a set of classes) that is integrated into a client application. It provides the client application with the ability to communicate with a specific web service via function calls. Client proxies allow the developer to work at a higher level, focusing on objects and patterns

rather than the manipulation and transport of raw XML. Most computing languages and platforms offer tools that take a WSDL file and associated XML schema files as input and automatically generate a client proxy for the developer.

The Messaging Web Service WSDL file can be found at <http://soap.m4u.com.au/?wsdl>. The Messaging Web Service implements a **document/literal wrapped** web service that conforms to WS-I Basic Profile 1.0. In general document/literal wrapped web services and in particular, WS-I Basic Profile compliant web services, provide the highest level of interoperability with client applications and third-party tools.

5.1.1 USING THE WSDL FILE TO GENERATING CLIENT CODE

This section briefly describes the steps necessary to generate a SOAP client capable of communicating with the MessageMedia Messaging Web Service. Only some of the more popular SOAP toolkits are covered here. All of these toolkits may be freely downloaded from their respective websites.

5.1.1.1 GENERATING A CLIENT PROXY FOR .NET

Microsoft provides two tools that may be used to generate client proxies capable of communicating with the MessageMedia Messaging Web Service: `wsdl.exe` and `svcutil.exe`. For the purposes of generating a .NET SOAP client for the Messaging Web Service the differences between these tools are subtle and either may be used to equal success. Both tools are capable of generating client proxies in a variety of .NET languages. The one you choose may depend on your development environment. `wsdl.exe` is described in detail [here](#) and `svcutil.exe` is described in detail [here](#). An article describing the difference between these two tools can be found [here](#).

Following is an example of how to generate a client proxy in C# using `wsdl.exe`. This command will generate the client proxy file `MessageMediaService.cs` in the local directory.

```
wsdl.exe /language:CS http://soap.m4u.com.au/?wsdl
```

Following is an example of how to generate a client proxy in VB .NET using `svcutil.exe`. This command will generate the client proxy file `MessageMediaService.vb` in the local directory.

```
SvcUtil.exe /language:VB http://soap.m4u.com.au/?wsdl
```

5.1.1.2 GENERATING A CLIENT PROXY FOR JAVA

The Java API for XML Web Services (JAX-WS) is provided by Sun Microsystems for creating web services in Java. It is part of the Java Enterprise Edition (EE) platform. More information on JAX-WS may be found [here](#).

Following is an example of how to generate a client proxy in Java. This command will generate both `.java` and `.class` files in the local directory.

```
wsimport.bat -keep http://soap.m4u.com.au/?wsdl
```

5.1.1.3 GSOAP FOR C/C++

gSOAP is an open source C and C++ software development toolkit for SOAP web services. It can be run on a variety of operating systems including Linux, Mac OS X and Windows. It can be used to produce client proxies in C and C++. More information about gSOAP can be found [here](#). The first step to creating a client proxy with gSOAP is to run the WSDL/Schema processor. Following is an example command which generates a C++ WSDL header file which is subsequently used to generate the client proxy.

```

wsdl2h -o MessageMediaService.h http://soap.m4u.com.au/?wsdl
http://xml.m4u.com.au/2009/CheckUser.xsd
http://xml.m4u.com.au/2009/CheckUserResponse.xsd
http://xml.m4u.com.au/2009/SendMessages.xsd
http://xml.m4u.com.au/2009/SendMessagesResponse.xsd
http://xml.m4u.com.au/2009/CheckReplies.xsd
http://xml.m4u.com.au/2009/CheckRepliesResponse.xsd
http://xml.m4u.com.au/2009/CheckReports.xsd
http://xml.m4u.com.au/2009/CheckReportsResponse.xsd
http://xml.m4u.com.au/2009/ConfirmReplies.xsd
http://xml.m4u.com.au/2009/ConfirmRepliesResponse.xsd
http://xml.m4u.com.au/2009/ConfirmReports.xsd
http://xml.m4u.com.au/2009/ConfirmReportsResponse.xsd
http://xml.m4u.com.au/2009/DeleteScheduledMessages.xsd
http://xml.m4u.com.au/2009/DeleteScheduledMessagesResponse.xsd
http://xml.m4u.com.au/2009/BlockNumbers.xsd
http://xml.m4u.com.au/2009/BlockNumbersResponse.xsd
http://xml.m4u.com.au/2009/UnblockNumbers.xsd
http://xml.m4u.com.au/2009/UnblockNumbersResponse.xsd
http://xml.m4u.com.au/2009/GetBlockedNumbers.xsd
http://xml.m4u.com.au/2009/GetBlockedNumbersResponse.xsd
http://xml.m4u.com.au/2009/FaultResponse.xsd

```

Note how all the schema files are also passed to `wsdl2h` on the command line. As the WSDL file references multiple schema files **gSOAP** requires that all schema files be specified explicitly. If they are not the generated header file will contain warnings.

The generated header file uses the STL vector class by default. This can be turned off by specifying the `-s` command line option. If STL vectors are used the `stlvector.h` file must be copied into the local development directory. It can be found in the `import` directory located in the directory in which **gSOAP** was installed.

The next step is to run the **gSOAP** code generator which creates the actual source files for the client proxy.

```
soapcpp2 MessageMediaService.h
```

This command will produce a number of source, header and XML files in the local directory. The XML files provide examples of the SOAP request and response formats employed by the MessageMedia Messaging Web Service. The files needed to build the client proxy are

```

soapMessageMediaServiceBindingProxy.h, MessageMediaServiceBinding.nsmmap,
soapH.h soapStub.h, stlvector.h, soapClient.cpp and soapC.cpp.

```

5.1.2 SOAP REQUEST FORMAT

The following listing provides an example of a Send Messages SOAP request.

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://xml.m4u.com.au/2009">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:sendMessages>
      <ns:authentication>
        <ns:userId>Username</ns:userId>
        <ns:password>Password</ns:password>
      </ns:authentication>
      <ns:requestBody>
        <ns:messages>
          <ns:message format="SMS" sequenceNumber="1">
            <ns:recipients>
              <ns:recipient uid="1">61400000001</ns:recipient>
            </ns:recipients>
          </ns:message>
        </ns:messages>
      </ns:requestBody>
    </ns:sendMessages>
  </soapenv:Body>
</soapenv:Envelope>

```

```

        <ns:recipient uid="2">6140000002</ns:recipient>
      </ns:recipients>
      <ns:content>Message Content</ns:content>
    </ns:message>
  </ns:messages>
</ns:requestBody>
</ns:sendMessages>
</soapenv:Body>
</soapenv:Envelope>

```

Listing 5.1.2.1: Example Send-Messages SOAP Request

All SOAP requests to the Messaging Web Service follow the same general format. The body of the SOAP envelope contains the entire request in a single element. As the Messaging Web Service follows the document/literal wrapped convention the body of the request is contained within a single element with the same name as the SOAP method that is being invoked, in this case `sendMessages`. All request elements must be specified as being within the namespace <http://xml.m4u.com.au/2009>.

5.1.3 SOAP RESPONSE FORMAT

The following listing provides an example of a Send Messages SOAP response.

```

<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>
    <sendMessagesResponse xmlns="http://xml.m4u.com.au/2009">
      <result sent="2" scheduled="0" failed="0">
        <accountDetails type="daily" creditLimit="500000"
          creditRemaining="499916"/>
      </result>
    </sendMessagesResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

Listing 5.1.3.1: Example Send-Messages SOAP Response

All (non-Fault) SOAP responses made by the Messaging Web Service follow the same general format. The body of the SOAP response envelope contains a single element that describes the response. Following the document/literal wrapped convention the response element has the same name as the SOAP method being invoked but with “Response” appended; in this case `sendMessagesResponse`. All response elements are specified as being within the namespace <http://xml.m4u.com.au/2009>.

5.1.4 SOAP FAULT FORMAT

SOAP faults are returned by the MessageMedia Messaging Web Service if the requested action cannot be fulfilled. Specifically, a SOAP fault will be returned given any of the following conditions:

1. The SOAP request is not well-formed
2. The body of the SOAP request is not valid against the request schema files
3. The user cannot be authenticated due to an invalid user ID or password
4. The user has exceeded their per-day message limit (only pertains to the Send Messages request)
- 5.

The following listing provides an example SOAP fault sent by the Messaging Web Service in response to an invalid user ID or password.

```

<soapenv:Envelope
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
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>
  <soapenv:Fault>
    <faultcode>soapenv:Client</faultcode>
    <faultstring>The userId or password is incorrect</faultstring>
    <detail>
      <faultResponse xmlns="http://xml.m4u.com.au/2009">
        <error code="authenticationFailed"/>
      </faultResponse>
    </detail>
  </soapenv:Fault>
</soapenv:Body>
</soapenv:Envelope>
```

Listing 5.1.4.1: Example SOAP Fault Response

In this example the `detail` element of the SOAP fault contains a XML description of the error that occurred. In this case the error code provided is `authenticationFailed`.

If the SOAP request envelope is not considered valid XML a SOAP fault will be returned by the web service. The returned `faultstring` element will describe the location of the XML parse error. The `detail` element of the returned fault will be empty to indicate that the invalid SOAP request was never considered by the web service.

If the SOAP request envelope is well-formed (i.e. valid XML) but the body of the request is not valid against any of the request schemas a SOAP fault will be returned by the web service. If the invalid SOAP request contained enough information for the web service to invoke one of the SOAP methods the `detail` element of the fault response will specify the `invalidDataFormat` error code.

If the SOAP request was found to be well-formed and the request was validated against one of the request schemas but the user ID or password was found to be incorrect the returned fault response will specify the `authenticationFailed` error code.

If a Send Messages request was made but the user has insufficient daily credits remaining to send all messages in the batch the returned fault response will specify the `perDayMessageLimit` error code. This indicates that no messages from the request have been sent.

Error codes returned by fault responses are described in detail in Section 7.1.

5.2 XML TRANSPORT VIA HTTP-POST

In addition to SOAP the MessageMedia Messaging Web Service is also available via HTTP-POST. For ease of implementation it is recommended that developers integrate the Messaging service as a SOAP web service; given the prominence of SOAP development frameworks and tools, this saves having to generate and parse raw XML documents. If, for whatever reason, SOAP is not a viable option the HTTP-POST method is made available and provides the same level of functionality.

The Messaging Web Service is available as a HTTP-POST web service at <http://xml.m4u.com.au>. The HTTP-POST method follows the exact same XML format as the SOAP method—instead of the XML request being contained within the body of a SOAP envelope it is passed to the Messaging Web Service as a HTTP-POST message. The body of the POST should contain a single parameter, “XMLDATA”. The value of the parameter is a URL-encoded text string that contains the XML request document.

Section 6: XML Validation

The MessageMedia Messaging Web Service uses XML schema documents to validate requests. This ensures that requests are well-formed and in the correct format. If XML requests do not pass validation the requests are discarded and a fault response with an error code of `invalidDataFormat` is returned (see Section 7.1 more details on fault responses).

XML schema documents are provided for all request types and responses in Appendix A. These documents provide a formal definition of how requests and responses are structured. SOAP toolkits will use these schema files in addition to the WSDL file to generate the client proxy. This ensures that the client application sends SOAP requests in the correct format; it also informs the client applications of the SOAP response format to expect in response to those requests.

Each request type has a corresponding XML schema file that is used to validate requests of that type. Following are the web locations of the XML schema documents used to validate requests:

- <http://xml.m4u.com.au/2009/CheckUser.xsd>
- <http://xml.m4u.com.au/2009/SendMessages.xsd>
- <http://xml.m4u.com.au/2009/CheckReplies.xsd>
- <http://xml.m4u.com.au/2009/CheckReports.xsd>
- <http://xml.m4u.com.au/2009/ConfirmReplies.xsd>
- <http://xml.m4u.com.au/2009/ConfirmReports.xsd>
- <http://xml.m4u.com.au/2009/DeleteScheduledMessages.xsd>
- <http://xml.m4u.com.au/2009/BlockNumbers.xsd>
- <http://xml.m4u.com.au/2009/UnblockNumbers.xsd>
- <http://xml.m4u.com.au/2009/GetBlockedNumbers.xsd>

Each response type also has a corresponding XML schema file. These schema files describe the responses that can be expected from the web service. Following are the web locations of the XML response schema documents:

- <http://xml.m4u.com.au/2009/CheckUserResponse.xsd>
- <http://xml.m4u.com.au/2009/SendMessagesResponse.xsd>
- <http://xml.m4u.com.au/2009/CheckRepliesResponse.xsd>
- <http://xml.m4u.com.au/2009/CheckReportsResponse.xsd>
- <http://xml.m4u.com.au/2009/ConfirmRepliesResponse.xsd>
- <http://xml.m4u.com.au/2009/ConfirmReportsResponse.xsd>
- <http://xml.m4u.com.au/2009/DeleteScheduledMessagesResponse.xsd>
- <http://xml.m4u.com.au/2009/BlockNumbersResponse.xsd>
- <http://xml.m4u.com.au/2009/UnblockNumbersResponse.xsd>
- <http://xml.m4u.com.au/2009/GetBlockedNumbersResponse.xsd>
- <http://xml.m4u.com.au/2009/FaultResponse.xsd>

There are many software libraries freely available that perform XML schema validation. There are also many websites that provide online XML validation against specified schemas. These resources should be utilised by developers to ensure smooth integration of the Messaging Web Service offered by MessageMedia.

For more information about XML schema see [here](#) and [here](#).

Section 7: Messaging Requests and Responses

This section describes the XML format of requests and responses used by the MessageMedia Messaging Web Service. There are six requests supported by the XML Interface: Check User, Send Messages, Check Replies, Confirm Replies, Check Reports and Confirm Reports.

7.1 FAULT RESPONSE FORMAT

Fault responses are returned by the Messaging Web Service when a request cannot be fulfilled. If the Messaging service is being used as a SOAP web service the fault response will be returned within the detail element of the SOAP fault envelope as described in Section 5.1.4. If the Messaging service is being used as a generic HTTP-POST web service the fault response will be returned verbatim as a XML document.

The following listing provides an example fault response that would be returned if the XML request was either badly formed or was not valid against the relevant schema.

```
<faultResponse xmlns="http://xml.m4u.com.au/2009">
  <error code="invalidDataFormat"/>
</faultResponse>
```

Listing 7.1.1: Example Fault Response

Fault responses specify an error code that indicates the reason as to why the fault occurred. The following table describes each of the fault error codes in detail.

Error Code	Description
authenticationFailed	The user ID or password was invalid.
invalidDataFormat	Either the request was not well-formed XML or the request was not valid against the relevant XML schema.
perDayMessageLimit	There were not enough daily message credits to fulfill the Send Messages request. If there is not enough credit to send <i>all</i> messages in the request no messages will be sent. This error code only pertains to the Send Messages request.

Table 7.1.1: Fault Response Error Codes

A fault response always implies that no action was performed on behalf of the request. A `perDayMessageLimit` fault response implies that no messages were sent. For example, if the user has 100 daily message credits remaining but attempts to send a batch of 101 messages a `perDayMessageLimit` fault response will be returned because the Messaging Web Service could not fulfill the request in its entirety.

7.2 CHECK USER REQUEST

The Check User request is used to authenticate a user and obtain their account credit details.

7.2.1 FORMAT OF THE CHECK USER REQUEST

Listing 7.2.1.1 shows an example Check User request.

```
<checkUser xmlns="http://xml.m4u.com.au/2009">
  <authentication>
```

```

    <userId>Username</userId>
    <password>Password</password>
</authentication>
</checkUser>

```

Listing 7.2.1.1: Example Check User Request

7.2.2 EXPLANATION OF THE CHECK USER REQUEST FORMAT

This section provides an explanation of the elements and attributes that are used in the Check User request. Element names are specified in bold and written as **<element>**. Attribute names are specified in bold and written as **attribute**. For a definitive specification of the Check User request see the Check User request XML schema provided in Appendix A.1.

<checkUser>	The root element of the Check User request.
xmlns	The XML namespace attribute. This value of this attribute must always be "http://xml.m4u.com.au/2009" as this is the target namespace.
<authentication>	This element contains userId and password elements which are required to authenticate the user.
<userId>	This element specifies the user ID which, in conjunction with the password, is required to authenticate the user. The user ID is the account name that the user is assigned by MessageMedia.

Table 7.2.2.1: Explanation of Check User Request Format

7.3 CHECK USER RESPONSE

The Check User response is returned by the MessageMedia Messaging Web Service in response to a valid, authenticated Check User request. The response provides the account credit details of the user.

7.3.1 FORMAT OF THE CHECK USER RESPONSE

Listing 7.3.1.1 shows an example Check User response.

```

<checkUserResponse xmlns="http://xml.m4u.com.au/2009">
  <result>
    <accountDetails type="daily" creditLimit="5000"
creditRemaining="4995"/>
  </result>
</checkUserResponse>

```

Listing 7.3.1.1: Example Check User Response

7.3.2 EXPLANATION OF THE CHECK USER RESPONSE FORMAT

This section provides an explanation of the elements and attributes that are used in the Check User response. Element names are specified in bold and written as **<element>**. Attribute names are specified in bold and written as **attribute**. For a definitive specification of the Check User response see the Check User response XML schema provided in Appendix A.2.

<checkUserResponse>	The root element of the Check User response.
xmlns	The XML namespace attribute. This value of this attribute is always "http://xml.m4u.com.au/2009" as this is the target namespace.
<result>	This element contains the result of the Check User request.

<accountDetails>	This element provides attributes which summarise the credit details of the account. This element will not be present if errors occur.
Type	This attribute specifies the type of crediting used by the account. The value of this attribute is always "daily" as only the daily crediting type exists. Account credit limits are per-day message limits and are refreshed each day.
creditLimit	This attribute specifies the daily credit limit of the account. This value indicates the number of individual SMS or voice messages which may be sent each day. For changes to this limit users should speak to their sales representative.
creditRemaining	This attribute specifies the amount of daily credit remaining for the account. This value indicates the number of individual SMS or voice messages which may be sent for the current day.

Table 7.3.2.1: Explanation of Check User Response Format

7.4 SEND MESSAGES REQUEST

The Send Messages request is used to send one or more SMS or voice messages to one or more recipients. The MessageMedia Messaging Web Service does not place a hard limit on the number of messages that may be placed in a request but users should be aware that it may be more efficient to split large batches of messages into multiple requests to avoid timing out their internet connections. In general, provided the user has a sufficient internet connection, batches of up to one thousand messages should be fine. Batches larger than this should be split up into multiple requests. The XML Interface allows two types of messages to be sent: SMS and voice. SMS messages may only be sent to mobile devices; voice messages, on the other hand, may be sent to landlines and mobile devices. Voice messages will be read out to the recipient by a text-to-speech software application. The list of messages in the Send Messages request may consist of both SMS and voice messages types and each message may have multiple recipients.

7.4.1 FORMAT OF THE SEND MESSAGES REQUEST

Listing 7.4.1.1 shows an example Send Messages request.

```
<sendMessages xmlns="http://xml.m4u.com.au/2009">
<authentication>
  <userId>Username</userId>
  <password>Password</password>
</authentication>
<requestBody>
  <messages sendMode="normal">
    <message format="SMS" sequenceNumber="1">
      <origin>61410000000</origin>
      <recipients>
        <recipient uid="1">61410000001</recipient>
        <recipient uid="2">61410000002</recipient>
        <recipient uid="3">61410000003</recipient>
        <recipient uid="4">61410000004</recipient>
      </recipients>
      <content>Message 1</content>
    </message>
    <message format="SMS" sequenceNumber="2">
      <recipients>
        <recipient uid="5">61410000005</recipient>
      </recipients>
      <scheduled>2012-12-25T15:30:00Z</scheduled>
      <content>Message 2</content>
    </message>
    <message format="voice" sequenceNumber="3">
      <recipients>
```

```

    <recipient uid="6">61410000006</recipient>
  </recipients>
  <content>Message 3</content>
</message>
<message format="SMS" sequenceNumber="4">
  <recipients>
    <recipient uid="7">61410000007</recipient>
    <recipient uid="8">61410000008</recipient>
  </recipients>
  <deliveryReport>true</deliveryReport>
  <validityPeriod>143</validityPeriod>
  <content>Message 4</content>
</message>
</messages>
</requestBody>
</sendMessages>

```

Listing 7.4.1.1: Example of a Send Messages Request

Listing 7.4.1.2 shows an example Send Messages request with Tags.

```

<sendMessages xmlns="http://xml.m4u.com.au/2009">
  <authentication>
    <userId>Username</userId>
    <password>Password</password>
  </authentication>
  <requestBody>
    <messages sendMode="normal">
      <message format="SMS" sequenceNumber="1">
        <recipients>
          <recipient uid="5">61410000005</recipient>
        </recipients>
        <content>Message 1</content>
        <tags>
          <tag name="costCode">101</tag>
          <tag name="dept">AAAA</tag>
        </tags>
      </message>
    </messages>
  </requestBody>
</sendMessages>

```

Listing 7.4.1.2: Example of a Send Messages Request with Tags.

7.4.2 EXPLANATION OF THE SEND MESSAGES REQUEST

This section provides an explanation of the elements and attributes that are used in the Send Messages request. Element names are specified in bold and written as **<element>**. Attribute names are specified in bold and written as **attribute**. For a definitive specification of the Send Messages request see the Send Messages request XML schema provided in Appendix A.3.

<sendMessages>	The root element of the Send Messages request.
xmlns	The XML namespace attribute. This value of this attribute must always be " http://xml.m4u.com.au/2009 " as this is the target namespace.
<authentication>	This element contains userId and password elements which are required to authenticate the user.

<userId>	This element specifies the user ID which, in conjunction with the password, is required to authenticate the user. The user ID is the account name that the user is assigned by MessageMedia.
<password>	This element specifies the password of the user which is used for authentication.
<requestBody>	This element contains the list of messages to send.
<messages>	The body of a Send Messages request must always contain this element. This element specifies all the messages that are requested to be sent.
sendMode	<p>(Optional) This attribute specifies a "send mode" to be used to send the messages. Normally, this attribute should not be set, or be set to the default value of "normal". Supported send modes are:</p> <ul style="list-style-type: none"> • "dropAll" – to drop (not send) the requested messages, and return a result indicating that messages were sent / scheduled successfully or failed to send at random. • "dropAllWithErrors" – to drop (not send) the requested messages, and return a result indicating that all messages failed to send. • "dropAllWithSuccess" – to drop (not send) the requested messages, but return a result indicating all messages were sent / scheduled successfully. • "normal" – to send the requested messages as normal. <p>The "dropAll*" modes are intended for testing purposes only.</p>
<message>	This element specifies a single message. A message always consists of one or more recipients and message content.
format	This attribute specifies the format of the message. The format must be either "SMS" or "voice". If a format other than these is specified the request will not pass schema validation and an <code>invalidDataFormat</code> error will be returned.
sequenceNumber	<p>(Optional) This attribute specifies a sequence number that is assigned to the message and is used to identify the message if an error occurs. Each message error in the response will specify the sequence number of the message that caused the error. Sequence numbers should be unique within the request.</p> <p>Sequence numbers must be unsigned integers and may range from 1 to 2147483647. If no sequence number is specified and a message error occurs the error will have a sequence number of zero.</p>
<origin>	<p>(Optional) This element specifies the message source address. The specified address will be used wherever possible, however due to limitations with various carriers, legislation etc, the final message is not guaranteed to come from the specified address.</p> <p>The address should not contain more than 11 characters, and should only consist of the following characters: 0-9, a-z, A-Z, and _</p> <p>This element will be ignored if the <code>scheduled</code> element is present.</p> <p>This element will be ignored if the corresponding feature has not been enabled in the MessageMedia gateway for this account.</p>
<recipients>	This element contains the one or more <code>recipient</code> elements.

<recipient>	<p>This element specifies a recipient of the message. A message must have one or more recipients.</p> <p>The recipient number may be specified in either international or non-international format. If the number is specified in non-international format then the international prefix of the user's operating country will be used. The operating country may be specified by the user when the user signs up to use the service and may be changed via the MyAccount website.</p>
uid	<p>(Optional) This attribute specifies a user-defined unique ID that is assigned to a message-recipient pair. The uid is an unsigned integer that uniquely identifies a message sent to a particular recipient.</p> <p>uid values are used for three things: to identify a message-recipient in the case of an error; to match a reply message to the sent message it is in response to; and to match a delivery report to the sent message it is in response to.</p> <p>If no uid value is specified a default value of zero is assigned.</p> <p>When using message tags, it is important to include a unique UID attribute for each recipient for all messages sent via a user account to ensure accurate reporting.</p> <p>When using message tags, a non-zero UID must be included for each recipient. Not including a UID or setting the UID to 0 will result in a validation error.</p>
<scheduled>	<p>(Optional) This element may optionally be used to schedule a message for future delivery. The content of this element specifies the date and time at which the message should be sent and must be specified in the standard XML schema dateTime format. The format of the dateTime data type is described in detail here and here.</p> <p>Briefly, the contents of this element should be specified in the format "YYYY-MM-DDThh:mm:ss" where:</p> <ul style="list-style-type: none"> • YYYY indicates the year • MM indicates the month • DD indicates the day • T indicates the start of the required time section • hh indicates the hour • mm indicates the minute • ss indicates the second <p>To avoid confusion scheduled messages should always specify a time zone. To specify a time zone you can either enter a dateTime in UTC time by appending a "Z" as in "2010-12-25T15:30:00Z" as is specified in the example in Listing 7.4.1. Time zones may also be specified as offsets from UTC by appending a positive or negative time as in "2010-12-25T15:30:00+05:00" or "2010-12-25T15:30:00-05:00".</p> <p>Messages that are scheduled for a date and time less than or equal to the current date and time will be sent immediately.</p>

<p><deliveryReport></p>	<p>(Optional) This element is a Boolean element which specifies whether delivery reporting is requested for the message. If this attribute is not specified a default value of "false" is assumed and no delivery reporting is requested. If this element has a value of "true" a delivery report will be requested for each message-recipient.</p> <p>Delivery reports only pertain to SMS messages (i.e. it is not possible to receive a delivery report for a voice message). See Section 7.8 for more information on delivery reports.</p>
<p><validityPeriod></p>	<p>(Optional) This element specifies the validity period of the message. Should the message-recipient be unavailable the service provider will continue to attempt delivery until the delivery period expires.</p> <p>Validity period only applies to SMS messages. If this element is specified the value must be an unsigned byte (i.e. an integer between 0 and 255 inclusive). The value of the validityPeriod element determines the validity time period as follows:</p> <ul style="list-style-type: none"> • For values ranging from 0 to 143 the time period is equal to: <i>(value + 1) x 5 minutes</i> • For values ranging from 144 to 167 the time period is equal to: <i>12 hours + (value - 143) x 30 minutes</i> • For values ranging from 168 to 196 the time period is equal to: <i>(value - 166) x 1 day</i> • For values ranging from 197 to 255 the time period is equal to: <i>(value - 192) x 1 week</i> <p>If this element is not specified a value of 169 is assumed which equates to 3 days.</p>

<p><tags></p>	<p>(Optional) This element specifies the arbitrary set of tags which are name/value pairs. The tags are configured as custom fields for individual message (not actually sent out as part of message text).</p> <p>An example of this is tagging messages with a cost code to facilitate cost reconciliation for messages sent from various departments within an organisation:</p> <pre><tags> <tag name="costCode">101</tag> <tag name="dept">AAAA</tag> </tags></pre> <p>If the message tags feature has not been enabled by MessageMedia for this account, usage of this element will result in validation error. Please contact support for more information.</p> <p>Including the same tag twice as below will result in a validation error</p> <pre><tags> <tag name="costCode">101</tag> <tag name="costCode">102</tag> </tags></pre> <p>Tag names are case insensitive. The following tag blocks are identical:</p> <pre><tags> <tag name="costCode">101</tag> </tags></pre> <pre><tags> <tag name="COSTCODE">101</tag> </tags></pre> <pre><tags> <tag name="costcode">101</tag> </tags></pre> <p>When using message tags, it is important to include a unique UID attribute for each recipient for all messages sent via a user account to ensure accurate reporting.</p> <p>When using message tags, a non-zero UID must be included for each recipient. Not including a UID or setting the UID to 0 will result in a validation error.</p>
<p><content></p>	<p>This element specifies the content of the message. There is no hard limit on the size of the content. SMS messages greater than 160 characters are not split up into multiple SMS messages—SMS concatenation is used so that they are delivered to the recipient as a single long message.</p>

Table 7.4.2.1: Explanation of Send Messages Request Format

7.5 SEND MESSAGES RESPONSE

The Send Messages response is returned by the MessageMedia Messaging Web Service in response to a valid, authenticated Send Messages request. The response provides a summary of the result of the request, the updated account credit details and a list of any messages that could not be sent due to errors.

7.5.1 FORMAT OF THE SEND MESSAGES RESPONSE

Listing 7.5.1.1 shows an example Send Messages response that does not contain any errors.

```
<sendMessagesResponse xmlns="http://xml.m4u.com.au/2009">
  <result sent="50" scheduled="10" failed="0">
    <accountDetails type="daily" creditLimit="5000" creditRemaining="2500"/>
  </result>
</sendMessagesResponse>
```

Listing 7.5.1.1: Example Send Messages Response

In the example response shown in Listing 7.5.1.1 the result (specified by the `result` element) indicates that 50 individual messages were sent, 10 individual messages were scheduled and 0 individual messages failed. The `accountDetails` element provides a summary of the account details after the request has been fulfilled. This information can be used to keep track of remaining account credit.

Listing 7.5.1.2 shows an example Send Messages response that reports errors.

```
<sendMessagesResponse xmlns="http://xml.m4u.com.au/2009">
  <result sent="25" scheduled="5" failed="6">
    <accountDetails type="daily" creditLimit="5000" creditRemaining="1500"/>
    <errors>
      <error code="emptyMessageContent" sequenceNumber="1">
        <recipients>
          <recipient uid="1">61410000001</recipient>
          <recipient uid="2">61410000002</recipient>
          <recipient uid="3">61410000003</recipient>
          <recipient uid="4">61410000004</recipient>
        </recipients>
      </error>
      <error code="recipientBlocked" sequenceNumber="2">
        <recipients>
          <recipient uid="5">61400000001</recipient>
        </recipients>
      </error>
      <error code="invalidRecipient" sequenceNumber="3">
        <recipients>
          <recipient uid="6">ABC</recipient>
        </recipients>
      </error>
    </errors>
  </result>
</sendMessagesResponse>
```

Listing 7.5.1.2: Example Send Messages Response Containing Errors

In the example response shown in Listing 7.5.1.2 the `result` element indicates that 25 individual messages were sent, 5 individual messages were scheduled and 6 individual messages failed. In this example 3 errors are reported affecting 6 recipients in total. Each error specifies an error code that defines the type of error and the sequence number of the message that caused the error. Each affected recipient specifies the same `uid` value that was assigned to the recipient in the request. There are 4 possible error codes. Each of these error codes is described in the following table.

Error Code	Description
<code>invalidRecipient</code>	One or more recipients were invalid.
<code>recipientBlocked</code>	One or more recipients were on the blocked list.
<code>emptyMessageContent</code>	The message content was empty.

other	An unknown error occurred. <code>error</code> elements with this code may contain an additional <code>content</code> element with a human-readable description of the error. Users who receive this type of error should contact MessageMedia Support.
-------	--

Table 7.5.1.1: Types of Message Errors

Each `error` element in the response may be linked to the `message` element in the request that caused the error via the sequence number (`sequenceNumber` attribute). Additionally, each affected `recipient` element in the response can be linked to the `recipient` element in the request via the unique ID (`uid` attribute) assigned to that message-recipient. The `uid` attribute may also be used to match a reply message to the sent message that the reply is in response to. For this reason it is recommended that sequence numbers be unique within a single Send Messages request and unique IDs be unique over the period of time within which a reply could be received. If a database is being used to store messages sent by the client application, a common practice is to use the integer-based primary key of the message as the unique ID. This guarantees that the ID will be unique and easily allows reply messages to be matched to sent messages.

7.5.2 EXPLANATION OF THE SEND MESSAGES RESPONSE

THIS SECTION PROVIDES AN EXPLANATION OF THE ELEMENTS AND ATTRIBUTES THAT ARE used in the Send Messages response. Element names are specified in bold and written as `<element>`. Attribute names are specified in bold and written as `attribute`. For a definitive specification of the Send Messages response see the Send Messages response XML schema provided in Appendix A.4.

<code><sendMessagesResponse></code>	The root element of the Send Messages response.
<code>xmlns</code>	The XML namespace attribute. This value of this attribute is always <code>"http://xml.m4u.com.au/2009"</code> as this is the target namespace.
<code><result></code>	This attributes of this element provide a summary of the result of the Send Messages request. It contains the <code>accountDetails</code> element as well as an <code>errors</code> element if any errors occurred.
<code>sent</code>	This attribute specifies the number of message-recipients whose messages were successfully processed for sending. For example, if 3 messages were submitted in the request and each message had 5 recipients this attribute will report a value of 15.
<code>scheduled</code>	This attribute specifies the number of message-recipients whose messages were successfully scheduled for future delivery. For example, if 3 scheduled messages were submitted in the request and each message had 5 recipients this attribute will report a value of 15. If the scheduled date and time specified in the request is less than or equal to the current date and time the message will be sent immediately and will be reported in the <code>sent</code> count not the <code>scheduled</code> count.
<code>failed</code>	This attribute is a Boolean attribute which specifies the number of message-recipients whose messages were not successfully processed. This number includes both failed non-scheduled and failed scheduled messages. If this number is greater than 0 the <code>errors</code> element will be present inside the <code>result</code> element and it will contain errors pertaining to each of the affected recipients.
<code><accountDetails></code>	This element provides attributes which summarise the credit details of the account after the request has been fulfilled. It will also contain an <code>errors</code> element if (and only if) the <code>failed</code> attribute reports a value greater than 0.

type	This attribute specifies the type of crediting used by the account. The value of this attribute is always "daily" as only the daily crediting type exists. Account credit limits are per-day message limits and are refreshed each day.
creditLimit	This attribute specifies the daily credit limit of the account. This value indicates the number of individual SMS or voice messages which may be sent each day. For changes to this limit users should speak to their sales representative.
creditRemaining	This attribute specifies the amount of daily credit remaining for the account. This value indicates the number of individual SMS or voice messages which may be sent for the current day.
<errors>	(Optional) This element contains errors that occurred during the processing of the request and will be present only if errors occurred. If this element is present it will contain one or more error elements. This element will only be present if the failed attribute of the parent result element reports a value greater than 0. The total number of affected recipients reported by all error elements will be equal to the value of the failed attribute.
<error>	This element reports an error that occurred in the processing of the request.
code	This attribute specifies the error code of the error. The error code defines the type of error and is one of the values specified in Table 7.5.1.1. The error codes that may result from a Send Messages request are: invalidRecipient, recipientBlocked, emptyMessageContent and other.
sequenceNumber	For a message error this attribute specifies the sequence number of the message that resulted in the error. If no sequence number was assigned to the message in the request the value of this attribute will be zero.
<content>	(Optional) This element is sometimes specified within the error element. It is used to report additional error content. It is only used when value of the code attribute is "other".
<recipients>	This element contains one or more recipient elements.
<recipient>	This element specifies the recipient who was affected by the error and as such did not receive their message.
uid	This attribute specifies the user-defined unique ID that was assigned to the message-recipient pair in the request. For this reason unique values should be used so that the affected message-recipient can be correctly identified. If the uid was not specified in the request the value of this attribute will be zero.

Table 7.5.2.1: Explanation of Send Messages Response Format

7.6 CHECK REPLIES REQUEST

The Check Replies request is used to download reply messages that are waiting on the gateway. Reply messages are downloaded for a specific user account. Reply messages will remain marked as unsent and will be downloaded each time the Check Replies request is made until they are confirmed by the user as having been received. See Section 7.10 for details on confirming replies.

7.6.1 FORMAT OF THE CHECK REPLIES REQUEST

Listing 7.6.1.1 shows an example Check Replies request.

```
<checkReplies xmlns="http://xml.m4u.com.au/2009">
```

```

<authentication>
  <userId>Username</userId>
  <password>Password</password>
</authentication>
<requestBody>
  <maximumReplies>100</maximumReplies>
</requestBody>
</checkReplies>

```

Listing 7.6.1.1: Example Check Replies Request

7.6.2 EXPLANATION OF THE CHECK REPLIES REQUEST FORMAT

This section provides an explanation of the elements and attributes that are used in the Check Replies request. Element names are specified in bold and written as **<element>**. Attribute names are specified in bold and written as **attribute**. For a definitive specification of the Check Replies request see the Check Replies request XML schema provided in Appendix A.5.

<checkReplies>	The root element of the Check Replies request.
xmlns	The XML namespace attribute. This value of this attribute must always be "http://xml.m4u.com.au/2009" as this is the target namespace.
<authentication>	This element contains userId and password elements which are required to authenticate the user.
<userId>	This element specifies the user ID which, in conjunction with the password, is required to authenticate the user. The user ID is the account name that the user is assigned by MessageMedia.
<password>	This element specifies the password of the user which is used for authentication.
<requestBody>	This element contains the optional maximumReplies element.
<maximumReplies>	(Optional) This element is used to specify the maximum number of replies to download in the response. If this element is not specified all waiting reply messages will be downloaded in the response.

Table 7.6.2.1: Explanation of Check Replies Request Format

7.7 CHECK REPLIES RESPONSE

The Check Replies response is returned by the MessageMedia Messaging Web Service in response to a valid, authenticated Check Replies request. The response contains a list of unconfirmed reply messages waiting on the gateway. If the **maximumReplies** element was specified in the request then the value of that element will determine the maximum number of replies present in the response. If no maximum was specified all unconfirmed replies will be present in the response.

7.7.1 FORMAT OF THE CHECK REPLIES RESPONSE

Listing 7.7.1.1 shows an example Check Replies response.

```

<checkRepliesResponse xmlns="http://xml.m4u.com.au/2009">
  <result returned="3" remaining="0">
    <replies>
      <reply format="SMS" uid="1" receiptId="13067831">
        <origin>61400000001</origin>
        <received>2010-12-25T16:35:21Z</received>
        <content>Reply Content 1</content>
      </reply>
      <reply format="SMS" uid="2" receiptId="13067832">
        <origin>61400000002</origin>

```

```

    <received>2010-12-25T16:35:22Z</received>
    <content>Reply Content 2</content>
  </reply>
  <reply format="SMS" uid="3" receiptId="13067833">
    <origin>61400000003</origin>
    <received>2010-12-25T16:35:23Z</received>
    <content>Reply Content 3</content>
  </reply>
</replies>
</result>
</checkRepliesResponse>

```

Listing 7.7.1.1: Example Check Replies Response

7.7.2 EXPLANATION OF THE CHECK REPLIES RESPONSE FORMAT

This section provides an explanation of the elements and attributes that are used in the Check Replies response. Element names are specified in bold and written as **<element>**. Attribute names are specified in bold and written as **attribute**. For a definitive specification of the Check Replies response see the Check Replies response XML schema provided in Appendix A.6.

<checkRepliesResponse>	The root element of the Check Replies request.
xmlns	The XML namespace attribute. This value of this attribute is always "http://xml.m4u.com.au/2009" as this is the target namespace.
<result>	This element provides attributes which summarise the result of the Check Replies request. It contains a list containing any downloaded reply messages.
returned	This attribute specifies the number of replies returned in the response.
remaining	This attribute specifies the number of replies remaining on the gateway for the user.
<replies>	This element contains any reply messages downloaded from the gateway. It will contain zero or more reply elements. The reply messages are listed in ascending order of their received time.
<reply>	(Optional) This element is used to specify a single reply message.
format	This attribute specifies the format of the reply message. As only SMS reply messages are supported the value of this attribute will always be "SMS".
uid	This attribute specifies the unique user-defined ID assigned to the message that this reply is in response to. The value of this ID allows client applications to match inbound reply messages to outbound messages. If no uid was assigned to the outbound message this attribute will have a value of zero.
receiptId	This attribute specifies a unique receipt ID. This ID is to be used by the client application when confirming receipt of the reply. Until replies are confirmed they will be marked as unsent and will be downloaded each time the Check Replies request is made. See Section 7.10 for details on confirming replies.
<origin>	This element specifies the phone number of the sender of the reply message.

<received>	<p>This attribute specifies the date and time at which the gateway received the reply message and is specified in the standard XML schema dateTime format. The format of the dateTime data type is described in detail here and here. The received date and time is always specified in UTC.</p> <p>Briefly, the content of this element is specified in the format "YYYY-MM-DDThh:mm:ssZ" where:</p> <ul style="list-style-type: none"> • YYYY indicates the year • MM indicates the month • DD indicates the day • T indicates the start of the required time section • hh indicates the hour • mm indicates the minute • ss indicates the second • Z indicates the UTC time zone
<content>	This element specifies the content of the reply message.

Table 7.7.2.1: Explanation of Check Replies Response Format

7.8 CHECK REPORTS REQUEST

The Check Reports request is used to download delivery reports that are waiting on the gateway. Delivery reports are downloaded for a specific user account. A delivery report reports the delivery status of a sent message. Delivery reports may only be obtained for SMS messages not voice messages and must be requested explicitly in the Send Messages request (Section 7.4). Delivery reports will remain marked as unsent and will be downloaded each time the Check Reports request is made until they are confirmed by the user as having been received. See Section 7.12 for details on confirming reports.

7.8.1 FORMAT OF THE CHECK REPORTS REQUEST

Listing 7.8.1.1 shows an example Check Reports request.

```
<checkReports xmlns="http://xml.m4u.com.au/2009">
  <authentication>
    <userId>Username</userId>
    <password>Password</password>
  </authentication>
  <requestBody>
    <maximumReports>100</maximumReports>
  </requestBody>
</checkReports>
```

Listing 7.8.1.1: Example Check Reports Request

7.8.2 EXPLANATION OF THE CHECK REPORTS REQUEST FORMAT

This section provides an explanation of the elements and attributes that are used in the Check Reports request. Element names are specified in bold and written as **<element>**. Attribute names are specified in bold and written as **attribute**. For a definitive specification of the Check Reports request see the Check Reports request XML schema provided in Appendix A.7.

< checkReports>	The root element of the Check Reports request.
xmlns	The XML namespace attribute. This value of this attribute must always be "http://xml.m4u.com.au/2009" as this is the target namespace.

<authentication>	This element contains <code>userId</code> and <code>password</code> elements which are required to authenticate the user.
<userId>	This element specifies the user ID which, in conjunction with the password, is required to authenticate the user. The user ID is the account name that the user is assigned by MessageMedia.
<password>	This element specifies the password of the user which is used for authentication.
<requestBody>	This element contains the optional <code>maximumReplies</code> element.
<maximumReports>	(Optional) This element is used to specify the maximum number of delivery reports to download in the response. If this element is not specified all waiting delivery reports will be downloaded in the response.

Table 7.8.2.1: Explanation of Check Reports Request Format

7.9 CHECK REPORTS RESPONSE

The Check Reports response is returned by the MessageMedia Messaging Web Service in response to a valid, authenticated Check Reports request. The response contains a list of unconfirmed delivery reports waiting on the gateway. If the `maximumReports` element was specified in the request then the value of that element will determine the maximum number of delivery reports present in the response. If no maximum was specified all unconfirmed delivery reports will be present in the response.

7.9.1 FORMAT OF THE CHECK REPORTS RESPONSE

Listing 7.9.1.1 shows an example Check Reports response.

```
<checkReportsResponse xmlns="http://xml.m4u.com.au/2009">
  <result returned="4" remaining="0">
    <reports>
      <report uid="1" receiptId="1351" status="delivered">
        <recipient>61400000001</recipient>
        <timestamp>2009-10-08T15:31:21Z</timestamp>
      </report>
      <report uid="2" receiptId="1352" status="delivered">
        <recipient>61400000002</recipient>
        <timestamp>2009-10-08T15:31:22Z</timestamp>
      </report>
      <report uid="3" receiptId="1353" status="pending">
        <recipient>61400000003</recipient>
        <timestamp>2009-10-08T15:31:23Z</timestamp>
      </report>
      <report uid="4" receiptId="1354" status="failed">
        <recipient>61400000004</recipient>
        <timestamp>2009-10-08T15:31:24Z</timestamp>
      </report>
    </reports>
  </result>
</checkReportsResponse>
```

Listing 7.9.1.1: Example Check Reports Response

Delivery reports indicate the delivery status of a previously sent SMS message. The `status` attribute or the `report` element defines this status. Table 7.9.1.1 describes the possible values that this attribute may assume and what each of these statuses indicate.

Value of the <code>status</code> Element	Description
--	-------------

delivered	The message was delivered to the recipient successfully. The <code>timestamp</code> element indicates the date and time (in UTC) that the message was delivered to the recipient's handset.
pending	The message is pending delivery. Some service providers send this delivery status when the message is delivered to their network and then send the <code>delivered</code> status when the message is delivered to the handset. For example, if the recipient's handset is switched off some providers will first send this status and then send the <code>delivered</code> status when the recipient's handset is turned on and the message is delivered. The <code>timestamp</code> element indicates the date and time (in UTC) that the message was delivered to the service provider's network.
failed	Delivery of the message failed. The reason for this may be that the message could not be delivered within the message's validity period (for example, the recipient's phone was switched off for an extended period).

Table 7.9.1.1: Delivery Status Descriptions

7.9.2 EXPLANATION OF THE CHECK REPORTS RESPONSE FORMAT

This section provides an explanation of the elements and attributes that are used in the Check Reports response. Element names are specified in bold and written as `<element>`. Attribute names are specified in bold and written as `attribute`. For a definitive specification of the Check Reports response see the Check Reports response XML schema provided in Appendix A.8.

<code><checkReportsResponse></code>	The root element of the Check Reports response.
<code>xmlns</code>	The XML namespace attribute. This value of this attribute is always "http://xml.m4u.com.au/2009" as this is the target namespace.
<code><result></code>	This element provides attributes which summarise the result of the Check Reports request. It contains a list containing any downloaded delivery reports.
<code>returned</code>	This attribute specifies the number of delivery reports returned in the response.
<code>remaining</code>	This attribute specifies the number of delivery reports remaining on the gateway for the user.
<code><reports></code>	This element contains any delivery report downloaded from the gateway. It will contain zero or more <code>report</code> elements. The delivery reports are listed in ascending order of their timestamp.
<code><report></code>	(Optional) This element is used to specify a single delivery report.
<code>uid</code>	This attribute specifies the unique user-defined ID assigned to the message that this delivery report is in relation to. The value of this ID allows client applications to match delivery reports to outbound messages. If no <code>uid</code> was assigned to the outbound message this attribute will have a value of zero.
<code>receiptId</code>	This attribute specifies a unique receipt ID. This ID is to be used by the client application when confirming receipt of the delivery report. Until delivery reports are confirmed they will be marked as unsent and will be downloaded each time the Check Reports request is made. See Section 7.12 for details on confirming delivery reports.

<p>status</p>	<p>This attribute specifies the status of the message that the delivery report is in relation to. The status of the message can be one of the following values:</p> <ul style="list-style-type: none"> • delivered • pending • failed • unknown <p>These values are described in Table 7.9.1.1.</p>
<p><recipient></p>	<p>This element specifies the phone number of the message recipient that the delivery report is in relation to.</p>
<p><timestamp></p>	<p>This attribute specifies a date and time in UTC. The delivery status of the message (i.e. the value of the <code>status</code> element) determines the meaning of this element. If the delivery status is equal to "delivered" then the timestamp element indicates the date and time at which the message was received on the recipient's handset. If the status is equal to "pending" then the timestamp is equal to the date and time at which the message was received by the recipient's service provider's network. If the status is equal to "failed" then the timestamp is equal to the date and time at which the message was deemed failed by the recipient's service provider.</p> <p>This element is specified in the standard XML schema <code>dateTime</code> format. The format of the <code>dateTime</code> data type is described in detail here and here. The value of this element is always specified in UTC.</p> <p>Briefly, the content of this element is specified in the format "YYYY-MM-DDThh:mm:ssZ" where:</p> <ul style="list-style-type: none"> • YYYY indicates the year • MM indicates the month • DD indicates the day • T indicates the start of the required time section • hh indicates the hour • mm indicates the minute • ss indicates the second • Z indicates the UTC time zone

Table 7.9.2.1: Explanation of Check Reports Response Format

7.10 CONFIRM REPLIES REQUEST

The Confirm Replies request is used to confirm the receipt of reply messages that were downloaded from the gateway. Replies that are unconfirmed will be downloaded each time a Check Replies request is made. When reply messages are confirmed they are marked as sent and will not be downloaded again. It is not possible for a user to confirm replies that do not belong to them.

Reply messages must be confirmed on an individual basis. Replies are specified by their receipt ID. This receipt ID is the same receipt ID that the reply message was assigned in the Check Replies response. The receipt ID is specified by the attribute `receiptId`. See Section 7.7 for details on the Check Replies response.

7.10.1 FORMAT OF THE CONFIRM REPLIES REQUEST

Listing 7.10.1.1 shows an example Confirm Replies request.

```

<confirmReplies xmlns="http://xml.m4u.com.au/2009">
  <authentication>
    <userId>Username</userId>
    <password>Password</password>
  </authentication>
  <requestBody>
    <replies>
      <reply receiptId="13067831"/>
      <reply receiptId="13067832"/>
      <reply receiptId="13067833"/>
      <reply receiptId="13067834"/>
      <reply receiptId="13067835"/>
    </replies>
  </requestBody>
</confirmReplies>

```

Listing 7.10.1.1: Example Confirm Replies Request

7.10.2 EXPLANATION OF THE CONFIRM REPLIES REQUEST FORMAT

This section provides an explanation of the elements and attributes that are used in the Confirm Replies request. Element names are specified in bold and written as **<element>**. Attribute names are specified in bold and written as **attribute**. For a definitive specification of the Confirm Replies request see the Confirm Replies request XML schema provided in Appendix A.9.

<confirmRepliesResponse>	The root element of the Confirm Replies request.
xmlns	The XML namespace attribute. This value of this attribute must always be "http://xml.m4u.com.au/2009" as this is the target namespace.
<authentication>	This element contains userId and password elements which are required to authenticate the user.
<userId>	This element specifies the user ID which, in conjunction with the password, is required to authenticate the user. The user ID is the account name that the user is assigned by MessageMedia.
<password>	This element specifies the password of the user which is used for authentication.
<requestBody>	This element contains the list of reply messages to confirm.
<replies>	This element contains one or more reply elements.
<reply>	This element is used to specify a single reply message to be confirmed as having been received. This element contains no content. It specifies the reply message to confirm via its receiptId attribute.
receiptId	This attribute specifies the receipt ID of the reply message to confirm. The receipt ID must be the same receipt ID that was provided to the client application in the Check Replies response. It is not possible for a user to confirm replies that do not belong to them. See Section 7.7 for details on the Check Replies response.

Table 7.10.2.1: Explanation of Confirm Replies Request Format

7.11 CONFIRM REPLIES RESPONSE

The Confirm Replies response is returned by the MessageMedia Messaging Web Service in response to a valid, authenticated Confirm Replies request. The response reports the number of reply messages that were confirmed.

7.11.1 FORMAT OF THE CONFIRM REPLIES RESPONSE

Listing 7.11.1.1 shows an example Confirm Replies response.

```
<confirmRepliesResponse xmlns="http://xml.m4u.com.au/2009">
  <result confirmed="5"/>
</confirmRepliesResponse>
```

Listing 7.11.1.1: Example Confirm Replies Response

7.11.2 EXPLANATION OF THE CONFIRM REPLIES RESPONSE FORMAT

This section provides an explanation of the elements and attributes that are used in the Confirm Replies response. Element names are specified in bold and written as **<element>**. Attribute names are specified in bold and written as **attribute**. For a definitive specification of the Confirm Replies response see the Confirm Replies response XML schema provided in Appendix A.4.

<code>< confirmRepliesResponse></code>	The root element of the Confirm Replies response.
<code><result></code>	This element provides an attribute which specifies the number of replies confirmed.
<code>confirmed</code>	This attribute specifies the total number of replies that were confirmed by the request. Only replies that belong to the user can be confirmed. If erroneous receipt IDs were specified in the request they will not be included in this total.

Table 7.11.2.1: Explanation of Confirm Replies Response Format

7.12 CONFIRM REPORTS REQUEST

The Confirm Reports request is used to confirm the receipt of delivery reports that were downloaded from the gateway. Delivery reports that are unconfirmed will be downloaded each time a Check Reports request is made. When delivery reports are confirmed they are marked as sent and will not be downloaded again. It is not possible for a user to confirm delivery reports that do not belong to them. Delivery reports must be confirmed on an individual basis. Delivery reports are specified by their receipt ID. This receipt ID is the same receipt ID that the delivery report was assigned in the Check Reports response. The receipt ID is specified by the attribute `receiptId`. See Section 7.9 for details on the Check Reports response.

7.12.1 FORMAT OF THE CONFIRM REPORTS REQUEST

Listing 7.12.1.1 shows an example Check Reports request.

```
<confirmReports xmlns="http://xml.m4u.com.au/2009">
<authentication>
  <userId>Username</userId>
  <password>Password</password>
</authentication>
<requestBody>
  <reports>
    <report receiptId="1351"/>
    <report receiptId="1352"/>
    <report receiptId="1353"/>
    <report receiptId="1354"/>
  </reports>
</requestBody>
</confirmReports>
```

```

    <report receiptId="1355"/>
  </reports>
</requestBody>
</confirmReports>

```

Listing 7.12.1.1: Example Confirm Reports Request

7.12.2 EXPLANATION OF THE CONFIRM REPORTS REQUEST FORMAT

This section provides an explanation of the elements and attributes that are used in the Confirm Reports request. Element names are specified in bold and written as `<element>`. Attribute names are specified in bold and written as `attribute`. For a definitive specification of the Confirm Reports request see the Confirm Reports request XML schema provided in Appendix A.11.

<code><confirmReports></code>	The root element of the Confirm Reports request.
<code>xmlns</code>	The XML namespace attribute. This value of this attribute must always be "http://xml.m4u.com.au/2009" as this is the target namespace.
<code><authentication></code>	This element contains <code>userId</code> and <code>password</code> elements which are required to authenticate the user.
<code><userId></code>	This element specifies the user ID which, in conjunction with the password, is required to authenticate the user. The user ID is the account name that the user is assigned by MessageMedia.
<code><password></code>	This element specifies the password of the user which is used for authentication.
<code><requestBody></code>	This element contains the list of delivery reports to confirm.
<code><reports></code>	This element contains one or more <code>report</code> elements.
<code><report></code>	This element is used to specify a single delivery report to be confirmed as having been received. This element contains no content. It specifies the delivery report to confirm via its <code>receiptId</code> attribute. This attribute specifies the receipt ID of the delivery report to confirm. The receipt ID must be the same receipt ID that was provided to the client application in the Check Reports response. It is not possible for a user to confirm delivery reports that do not belong to them. See Section 7.9 for details on the Check Reports response.
<code>receiptId</code>	

Table 7.12.2.1: Explanation of Confirm Reports Request Format

7.13 CONFIRM REPORTS RESPONSE

The Confirm Reports response is returned by the MessageMedia Messaging Web Service in response to a valid, authenticated Confirm Reports request. The response reports the number of delivery reports that were confirmed.

7.13.1 FORMAT OF THE CONFIRM REPORTS RESPONSE

Listing 7.13.1.1 shows an example Confirm Reports response.

```

<confirmReportsResponse xmlns="http://xml.m4u.com.au/2009">
  <result confirmed="5"/>
</confirmReportsResponse>

```

Listing 7.13.1.1: Example Confirm Reports Response

7.13.2 EXPLANATION OF THE CONFIRM REPLIES RESPONSE FORMAT

This section provides an explanation of the elements and attributes that are used in the Confirm Reports response. Element names are specified in bold and written as **<element>**. Attribute names are specified in bold and written as **attribute**. For a definitive specification of the Confirm Reports response see the Confirm Reports response XML schema provided in Appendix A.12.

<code><confirmReportsResponse></code>	The root element of the Confirm Reports response.
<code>xmlns</code>	The XML namespace attribute. This value of this attribute is always "http://xml.m4u.com.au/2009" as this is the target namespace.
<code>< result></code>	This element provides an attribute which specifies the number of delivery reports confirmed.
<code>confirmed</code>	This attribute specifies the total number of delivery reports that were confirmed by the request. Only reports that belong to the user can be confirmed. If erroneous receipt IDs were specified in the request they will not be included in this total.

Table 7.13.2.1: Explanation of Confirm Reports Response Format

7.14 DELETE SCHEDULED MESSAGES REQUEST

The Delete Scheduled Messages request is used to request the uncheduling of messages that have been submitted to the gateway but are still yet to be sent. Only messages that were given a `scheduled` timestamp in the Send Messages request can be uncheduled. Only messages sent from the given account can be uncheduled. Messages submitted to the gateway via other APIs may be deleted via this method.

Messages must be confirmed on an individual basis. Messages are specified by their message ID. This message ID is the same message ID that was specified in recipient `uid` attribute in the Send Messages request. Messages with an unrecognised message ID will be ignored. See Section 7.4 for details on the Send Messages request.

7.14.1 FORMAT OF THE DELETE SCHEDULED MESSAGES REQUEST

Listing 7.14.1.1 shows an example Delete Scheduled Messages request.

```
<deleteScheduledMessages xmlns="http://xml.m4u.com.au/2009">
<authentication>
  <userId>Username</userId>
  <password>Password</password>
</authentication>
<requestBody>
  <messages>
    <message messageId="1351"/>
    <message messageId="1353"/>
    <message messageId="1354"/>
  </messages>
</requestBody>
</deleteScheduledMessages>
```

Listing 7.14.1.1: Example Confirm Reports Request

7.14.2 EXPLANATION OF THE DELETE SCHEDULED MESSAGES REQUEST FORMAT

This section provides an explanation of the elements and attributes that are used in the Delete Scheduled Messages request. Element names are specified in bold and written as **<element>**. Attribute names are specified in bold and written as **attribute**. For a definitive specification of the Delete Scheduled Messages request see the Delete Scheduled Messages request XML schema provided in Appendix A.13.

<deleteScheduledMessages>	The root element of the Delete Scheduled Messages request.
xmlns	The XML namespace attribute. This value of this attribute must always be "http://xml.m4u.com.au/2009" as this is the target namespace.
<authentication>	This element contains <code>userId</code> and <code>password</code> elements which are required to authenticate the user.
<userId>	This element specifies the user ID which, in conjunction with the password, is required to authenticate the user. The user ID is the account name that the user is assigned by MessageMedia.
<password>	This element specifies the password of the user which is used for authentication.
<requestBody>	This element contains the list of messages to unschedule.
<messages>	This element contains one or more <code>message</code> elements.
<message>	This element is used to specify a single message to be deleted from the scheduled send queue. This element contains no content. It specifies the message to delete via its <code>messageId</code> attribute.
messageId	This attribute specifies the message ID of the message to unschedule. The message ID must be the same message ID that was specified in the Send Messages request. It is not possible for a user to delete messages that do not belong to them. See Section 7.4 for details on the Send Messages request.

Table 7.14.2.1: Explanation of Delete Scheduled Messages s Request Format

7.15 DELETE SCHEDULED MESSAGES RESPONSE

The Delete Scheduled Messages response is returned by the MessageMedia Messaging Web Service in response to a valid, authenticated Delete Scheduled Messages request. The response reports the number of messages that were unscheduled.

7.15.1 FORMAT OF THE DELETE SCHEDULED MESSAGES RESPONSE

Listing 7.15.1.1 shows an example Delete Scheduled Messages response.

```
<deleteScheduledMessagesResponse xmlns="http://xml.m4u.com.au/2009">
  <result unscheduled="3"/>
</deleteScheduledMessagesResponse>
```

Listing 7.15.1.1: Example Delete Scheduled Messages Response

7.15.2 EXPLANATION OF THE DELETE SCHEDULED MESSAGES RESPONSE FORMAT

This section provides an explanation of the elements and attributes that are used in the Delete Scheduled Messages response. Element names are specified in bold and written as **<element>**. Attribute names are specified in bold and written as **attribute**. For a definitive specification of the Delete Scheduled Messages response see the Delete Scheduled Messages response XML schema provided in Appendix A.14.

<deleteScheduledMessagesResponse>	The root element of the Delete Scheduled Messages response.
-----------------------------------	---

Xmlns	The XML namespace attribute. This value of this attribute is always "http://xml.m4u.com.au/2009" as this is the target namespace.
<result>	This element provides an attribute which specifies the number of messages deleted.
unscheduled	This attribute specifies the total number of messages that were deleted by the request. Only messages that belong to the user can be deleted. If erroneous messageIDs were specified in the request they will not be included in this total.

Table 7.15.2.1: Explanation of Delete Scheduled Messages Response Format

7.16 BLOCK NUMBERS REQUEST

The Block Numbers request is used to prevent the authenticated account being able to send messages to the specified numbers in future.

7.16.1 FORMAT OF THE BLOCK NUMBERS REQUEST

Listing 7.16.1.1 shows an example Block Numbers request.

```
<blockNumbers xmlns="http://xml.m4u.com.au/2009">
<authentication>
  <userId>Username</userId>
  <password>Password</password>
</authentication>
<requestBody>
  <recipients>
    <recipient uid="1">6141000001</recipient>
    <recipient uid="2">6141000002</recipient>
    <recipient uid="3">6141000003</recipient>
    <recipient uid="4">6141000004</recipient>
  </recipients>
</requestBody>
</blockNumbers>
```

Listing 7.16.1.1: Example Block Numbers Request

7.16.2 EXPLANATION OF THE BLOCK NUMBERS REQUEST FORMAT

This section provides an explanation of the elements and attributes that are used in the Block Numbers request. Element names are specified in bold and written as **<element>**. Attribute names are specified in bold and written as **attribute**. For a definitive specification of the Block Numbers request see the Block Numbers request XML schema provided in Appendix A.15.

<blockNumbers>	The root element of the Block Numbers request.
xmlns	The XML namespace attribute. This value of this attribute is always "http://xml.m4u.com.au/2009" as this is the target namespace.
<authentication>	This element contains userId and password elements which are required to authenticate the user.
<userId>	This element specifies the user ID which, in conjunction with the password, is required to authenticate the user. The user ID is the account name that the user is assigned by MessageMedia.
<password>	This element specifies the password of the user which is used for authentication.

<requestBody>	This element contains the list of recipients to block.
<recipients>	This element contains the one or more <i>recipient</i> elements.
<recipient>	This element specifies a recipient block.
uid	(Optional) This attribute specifies a user-defined unique ID that is assigned to the recipient. The UID is an unsigned integer that uniquely identifies the recipient for a given Block Numbers request. uid values are used to report which recipients failed to be blocked in the Block Numbers response. If no uid value is specified a default value of zero is assigned.

Table 7.16.2.1: Explanation of Block Numbers Request Format

7.17 BLOCK NUMBERS RESPONSE

The Block Numbers response is returned by the MessageMedia Messaging Web Service in response to a valid, authenticated Block Numbers request. The response provides a summary of the result of the request, including a list of any recipients that could not be blocked.

7.17.1 FORMAT OF THE BLOCK NUMBERS RESPONSE

Listing 7.17.1.1 shows an example Block Numbers response.

```
<blockNumbersResponse xmlns="http://xml.m4u.com.au/2009">
  <result blocked="2" failed="2">
    <errors>
      <error code="invalidRecipient" sequenceNumber="0">
        <recipients>
          <recipient uid="1">6141000001</recipient>
          <recipient uid="2">6141000002</recipient>
        </recipients>
      </error>
    </errors>
  </result>
</blockNumbersResponse>
```

Listing 7.17.1.1: Example Block Numbers Response

7.17.2 EXPLANATION OF THE BLOCK NUMBERS RESPONSE FORMAT

This section provides an explanation of the elements and attributes that are used in the Block Numbers response. Element names are specified in bold and written as **<element>**. Attribute names are specified in bold and written as **attribute**. For a definitive specification of the Block Numbers response see the Block Numbers response XML schema provided in Appendix A.16.

<blockNumbersResponse>	The root element of the Block Numbers response.
xmlns	The XML namespace attribute. This value of this attribute is always "http://xml.m4u.com.au/2009" as this is the target namespace.
<result>	This attributes of this element provide a summary of the result of the Send Messages request. It contains the <i>accountDetails</i> element as well as an <i>errors</i> element if any errors occurred.
blocked	This attribute specifies the number of recipients that were successfully added to the block list.

failed	This attribute specifies the number of recipients that could not be added to the block list. If this number is greater than 0 the <code>errors</code> element will be present inside the <code>result</code> element and it will contain errors pertaining to each of the affected recipients.
<errors>	(Optional) This element contains errors that occurred during the processing of the request and will be present only if errors occurred. If this element is present it will contain one or more <code>error</code> elements. This element will only be present if the <code>failed</code> attribute of the parent <code>result</code> element reports a value greater than 0. The total number of affected recipients reported by all <code>error</code> elements will be equal to the value of the <code>failed</code> attribute.
<error>	This element reports an error that occurred in the processing of the request.
code	This attribute specifies the error code of the error. The error code defines the type of error and is one of the values specified in Table 7.5.1.1. The error codes that may result from a Block Numbers request are: <code>invalidRecipient</code> and <code>other</code> .
sequenceNumber	This attribute attribute is not used, and will always be zero.
<recipients>	This element contains one or more <code>recipient</code> elements.
<recipient>	This element specifies the recipient who was affected by the error and as such was not blocked.
uid	This attribute specifies the user-defined unique ID that was assigned to the recipient in the request. For this reason unique values should be used so that the affected message-recipient can be correctly identified. If the <code>uid</code> was not specified in the request the value of this attribute will be zero.

Table 7.17.2.1: Explanation of Block Numbers Response Format

7.18 UNBLOCK NUMBERS REQUEST

The Unblock Numbers request is used to remove existing number blocks.

7.18.1 FORMAT OF THE UNBLOCK NUMBERS REQUEST

Listing 7.18.1.1 shows an example Unblock Numbers request.

```
<unblockNumbers xmlns="http://xml.m4u.com.au/2009">
  <authentication>
    <userId>Username</userId>
    <password>Password</password>
  </authentication>
  <requestBody>
    <recipients>
      <recipient uid="1">61410000001</recipient>
      <recipient uid="2">61410000002</recipient>
      <recipient uid="3">61410000003</recipient>
      <recipient uid="4">61410000004</recipient>
    </recipients>
  </requestBody>
</unblockNumbers>
```

Listing 7.18.1.1: Example Unblock Numbers Request

7.18.2 EXPLANATION OF THE UNBLOCK NUMBERS REQUEST FORMAT

This section provides an explanation of the elements and attributes that are used in the Unblock Numbers request. Element names are specified in bold and written as `<element>`. Attribute names are specified in bold and written as `attribute`. For a definitive specification of the Unblock Numbers request see the Unblock Numbers request XML schema provided in Appendix A.17.

<code><unblockNumbers></code>	The root element of the Unblock Numbers request.
<code>xmlns</code>	The XML namespace attribute. This value of this attribute is always <code>"http://xml.m4u.com.au/2009"</code> as this is the target namespace.
<code><authentication></code>	This element contains <code>userId</code> and <code>password</code> elements which are required to authenticate the user.
<code><userId></code>	This element specifies the user ID which, in conjunction with the password, is required to authenticate the user. The user ID is the account name that the user is assigned by MessageMedia.
<code><password></code>	This element specifies the password of the user which is used for authentication.
<code><requestBody></code>	This element contains the list of recipients to unblock.
<code><recipients></code>	This element contains the one or more <code>recipient</code> elements.
<code><recipient></code>	This element specifies a recipient unblock.
<code>uid</code>	(Optional) This attribute specifies a user-defined unique ID that is assigned to the recipient. The UID is an unsigned integer the uniquely identifies the recipient for a given Block Numbers request. <code>uid</code> values are used to report which recipients failed to be unblocked in the Unblock Numbers response. If no <code>uid</code> value is specified a default value of zero is assigned.

Table 7.18.2.1: Explanation of Unblock Numbers request Format

7.19 UNBLOCK NUMBERS RESPONSE

The Unblock Numbers response is returned by the MessageMedia Messaging Web Service in response to a valid, authenticated Unblock Numbers request. The response provides a summary of the result of the request, including a list of any recipients that could not be unblocked.

7.19.1 FORMAT OF THE UNBLOCK NUMBERS RESPONSE

Listing 7.19.1.1 shows an example Unblock Numbers response.

```
<unblockNumbersResponse xmlns="http://xml.m4u.com.au/2009">
  <result unblocked="2" failed="2">
    <errors>
      <error code="invalidRecipient" sequenceNumber="0">
        <recipients>
          <recipient uid="1">61410000001</recipient>
          <recipient uid="2">61410000002</recipient>
        </recipients>
      </error>
    </errors>
  </result>
</unblockNumbersResponse>
```

Listing 7.19.1.1: Example Unblock Numbers Response

7.19.2 EXPLANATION OF THE UNBLOCK NUMBERS RESPONSE FORMAT

This section provides an explanation of the elements and attributes that are used in the Unblock Numbers response. Element names are specified in bold and written as `<element>`. Attribute names are specified in bold and written as `attribute`. For a definitive specification of the Unblock Numbers response see the Unblock Numbers response XML schema provided in Appendix A.18.

<code><unblockNumbersResponse></code>	The root element of the Unblock Numbers response.
<code>xmlns</code>	The XML namespace attribute. This value of this attribute is always "http://xml.m4u.com.au/2009" as this is the target namespace.
<code><result></code>	This attributes of this element provide a summary of the result of the Send Messages request. It contains the <code>accountDetails</code> element as well as an <code>errors</code> element if any errors occurred.
<code>unblocked</code>	This attribute specifies the number of recipients that were successfully added to the block list.
<code>failed</code>	This attribute specifies the number of recipients that could not be removed from the block list. If this number is greater than 0 the <code>errors</code> element will be present inside the <code>result</code> element and it will contain errors pertaining to each of the affected recipients.
<code><errors></code>	(Optional) This element contains errors that occurred during the processing of the request and will be present only if errors occurred. If this element is present it will contain one or more <code>error</code> elements. This element will only be present if the <code>failed</code> attribute of the parent <code>result</code> element reports a value greater than 0. The total number of affected recipients reported by all <code>error</code> elements will be equal to the value of the <code>failed</code> attribute.
<code><error></code>	This element reports an error that occurred in the processing of the request.
<code>code</code>	This attribute specifies the error code of the error. The error code defines the type of error and is one of the values specified in Table 7.5.1.1. The error codes that may result from an Unblock Numbers request are: <code>invalidRecipient</code> and <code>other</code> .
<code>sequenceNumber</code>	This attribute attribute is not used, and will always be zero.
<code><recipients></code>	This element contains one or more <code>recipient</code> elements.
<code><recipient></code>	This element specifies the recipient who was affected by the error and as such was not unblocked.
<code>uid</code>	This attribute specifies the user-defined unique ID that was assigned to the recipient in the request. For this reason unique values should be used so that the affected message-recipient can be correctly identified. If the <code>uid</code> was not specified in the request the value of this attribute will be zero.

Table 7.19.2.1: Explanation of Unblock Numbers Response Format

7.20 GET BLOCKED NUMBERS REQUEST

The Get Blocked Numbers request is used retrieve a list of numbers that are currently blocked for the authenticated account.

7.20.1 FORMAT OF THE GET BLOCKED NUMBERS REQUEST

Listing 7.20.1.1 shows an example Get Blocked Numbers request.

```
<getBlockedNumbers xmlns="http://xml.m4u.com.au/2009">
  <authentication>
    <userId>Username</userId>
    <password>Password</password>
  </authentication>
  <requestBody>
    <maximumRecipients>50</maximumRecipients>
  </requestBody>
</getBlockedNumbers>
```

Listing 7.20.1.1: Example Get Blocked Numbers Request

7.20.2 EXPLANATION OF THE GET BLOCKED NUMBERS REQUEST FORMAT

This section provides an explanation of the elements and attributes that are used in the Get Blocked Numbers request. Element names are specified in bold and written as **<element>**. Attribute names are specified in bold and written as **attribute**. For a definitive specification of the Get Blocked Numbers request see the Get Blocked Numbers request XML schema provided in Appendix A.19.

<getBlockedNumbers>	The root element of the Get Blocked Numbers request.
xmlns	The XML namespace attribute. This value of this attribute is always "http://xml.m4u.com.au/2009" as this is the target namespace.
<authentication>	This element contains userId and password elements which are required to authenticate the user.
<userId>	This element specifies the user ID which, in conjunction with the password, is required to authenticate the user. The user ID is the account name that the user is assigned by MessageMedia.
<password>	This element specifies the password of the user which is used for authentication.
<requestBody>	This element contains the list of recipients to unblock.
<maximumRecipients>	(Optional) This element is used to specify the maximum number of blocked numbers to include in the response. If this element is not specified all blocked numbers will be included in the response.

Table 7.20.2.1: Explanation of Get Blocked Numbers Request Format

7.21 GET BLOCKED NUMBERS RESPONSE

The Get Blocked Numbers response is returned by the MessageMedia Messaging Web Service in response to a valid, authenticated Get Blocked Numbers request. The response provides a list of numbers currently blocked for the authenticated account.

7.21.1 FORMAT OF THE GET BLOCKED NUMBERS RESPONSE

Listing 7.21.1.1 shows an example Get Blocked Numbers response.

```
<getBlockedNumbersResponse xmlns="http://xml.m4u.com.au/2009">
  <result found="4" returned="4">
    <recipients>
      <recipient uid="0">61410000001</recipient>
      <recipient uid="0">61410000002</recipient>
      <recipient uid="0">61410000003</recipient>
      <recipient uid="0">61410000004</recipient>
    </recipients>
  </result>
</getBlockedNumbersResponse>
```

```
</result>
</getBlockedNumbersResponse>
```

Listing 7.21.1.1: Example Get Blocked Numbers Response

7.21.2 EXPLANATION OF THE GET BLOCKED NUMBERS RESPONSE FORMAT

This section provides an explanation of the elements and attributes that are used in the Get Blocked Numbers response. Element names are specified in bold and written as **<element>**. Attribute names are specified in bold and written as **attribute**. For a definitive specification of the Get Blocked Numbers response see the Get Blocked Numbers response XML schema provided in Appendix A.20.

<code><getBlockedNumbersResponse></code>	The root element of the Get Blocked Numbers response.
<code>xmlns</code>	The XML namespace attribute. This value of this attribute is always "http://xml.m4u.com.au/2009" as this is the target namespace.
<code><result></code>	This attributes of this element provide a summary of the result of the Send Messages request. It contains the <code>accountDetails</code> element as well as an <code>errors</code> element if any errors occurred.
<code>found</code>	This attribute specifies the number of recipients that were found to be blocked for the authenticated account.
<code>returned</code>	This attribute specifies the number of recipients that are included in this response, that is in the <code>recipients</code> element. This number will be the the smallest of the <code>found</code> attribute value and the <code>maximumRecipients</code> value (if any) specified in the Get Blocked Numbers request that this result is in response to.
<code><recipients></code>	This element contains one or more <code>recipient</code> elements.
<code><recipient></code>	This element specifies the recipient that is on the block list.
<code>uid</code>	This attribute attribute is not used, and will always be zero.

Table 7.21.2.1: Explanation of Get Blocked Numbers Response Format

Appendix A: XML Schema Files

This appendix provides a listing of each of the XML schema files that are used by the MessageMedia Messaging web service.

A.1 CHECK USER REQUEST SCHEMA

Shown in the following listing is CheckUser.xsd which can be found online at <http://xml.m4u.com.au/2009/CheckUser.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="RequestTypes.xsd"/>

  <xsd:element name="checkUser" type="CheckUserRequestType"/>

</xsd:schema>
```

Listing A.1.1: Check User Request XML Schema

A.2 CHECK USER RESPONSE SCHEMA

Shown in the following listing is CheckUserResponse.xsd which can be found online at <http://xml.m4u.com.au/2009/CheckUserResponse.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="ResponseTypes.xsd"/>

  <xsd:element name="checkUserResponse" type="CheckUserResponseType"/>

</xsd:schema>
```

Listing A.2.1: Check User Response XML Schema

A.3 SEND MESSAGES REQUEST SCHEMA

Shown in the following listing is SendMessages.xsd which can be found online at <http://xml.m4u.com.au/2009/SendMessages.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="RequestTypes.xsd"/>

  <xsd:element name="sendMessages" type="SendMessagesRequestType"/>

</xsd:schema>
```

Listing A.3.1: Send Messages Request XML Schema

A.4 SEND MESSAGES RESPONSE SCHEMA

Shown in the following listing is SendMessagesResponse.xsd which can be found online at <http://xml.m4u.com.au/2009/SendMessagesResponse.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="ResponseTypes.xsd"/>

  <xsd:element name="sendMessagesResponse"
  type="SendMessagesResponseType"/>

</xsd:schema>
```

Listing A.4.1: Send Messages Response XML Schema

A.5 CHECK REPLIES REQUEST SCHEMA

Shown in the following listing is CheckReplies.xsd which can be found online at <http://xml.m4u.com.au/2009/CheckReplies.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="RequestTypes.xsd"/>

  <xsd:element name="checkReports" type="CheckReportsRequestType"/>

</xsd:schema>
```

Listing A.5.1: Check Replies Request XML Schema

A.6 CHECK REPLIES RESPONSE SCHEMA

Shown in the following listing is CheckRepliesResponse.xsd which can be found online at <http://xml.m4u.com.au/2009/CheckRepliesResponse.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="ResponseTypes.xsd"/>

  <xsd:element name="checkRepliesResponse"
  type="CheckRepliesResponseType"/>

</xsd:schema>
```

Listing A.6.1: Check Replies Response XML Schema

A.7 CHECK REPORTS REQUEST SCHEMA

Shown in the following listing is CheckReports.xsd which can be found online at <http://xml.m4u.com.au/2009/CheckReports.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="RequestTypes.xsd"/>

  <xsd:element name="checkReports" type="CheckReportsRequestType"/>

</xsd:schema>
```

Listing A.7.1: Check Reports Request XML Schema

A.8 CHECK REPORTS RESPONSE SCHEMA

Shown in the following listing is CheckReportsResponse.xsd which can be found online at <http://xml.m4u.com.au/2009/CheckReportsResponse.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="ResponseTypes.xsd"/>

  <xsd:element name="checkReportsResponse"
  type="CheckReportsResponseType"/>

</xsd:schema>
```

Listing A.8.1: Check Reports Response XML Schema

A.9 CONFIRM REPLIES REQUEST SCHEMA

Shown in the following listing is ConfirmReplies.xsd which can be found online at <http://xml.m4u.com.au/2009/ConfirmReplies.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="RequestTypes.xsd"/>

  <xsd:element name="confirmReplies" type="ConfirmRepliesRequestType"/>

</xsd:schema>
```

Listing A.9.1: Confirm Replies Request XML Schema

A.10 CONFIRM REPLIES RESPONSE SCHEMA

Shown in the following listing is ConfirmRepliesResponse.xsd which can be found online at <http://xml.m4u.com.au/2009/ConfirmRepliesResponse.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="ResponseTypes.xsd"/>

  <xsd:element name="confirmRepliesResponse"
  type="ConfirmRepliesResponseType"/>

</xsd:schema>
```

Listing A.10.1: Confirm Replies Response XML Schema

A.11 CONFIRM REPORTS REQUEST SCHEMA

Shown in the following listing is ConfirmReports.xsd which can be found online at <http://xml.m4u.com.au/2009/ConfirmReports.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="RequestTypes.xsd"/>

  <xsd:element name="confirmReports" type="ConfirmReportsRequestType"/>

</xsd:schema>
```

Listing A.11.1: Confirm Reports Request XML Schema

A.12 CONFIRM REPORTS RESPONSE SCHEMA

Shown in the following listing is `ConfirmReportsResponse.xsd` which can be found online at <http://xml.m4u.com.au/2009/ConfirmReportsResponse.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="ResponseTypes.xsd"/>

  <xsd:element name="confirmReportsResponse"
    type="ConfirmReportsResponseType"/>
</xsd:schema>
```

Listing A.12.1: Confirm Reports Response XML Schema

A.13 DELETE SCHEDULED MESSAGES REQUEST SCHEMA

Shown in the following listing is `DeleteScheduledMessages.xsd` which can be found online at <http://xml.m4u.com.au/2009/DeleteScheduledMessages.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="RequestTypes.xsd"/>

  <xsd:element name="deleteScheduledMessages"
    type="DeleteScheduledMessagesRequestType"/>
</xsd:schema>
```

Listing A.17.1: Delete Scheduled Messages Request XML Schema

A.14 DELETE SCHEDULED MESSAGES RESPONSE SCHEMA

Shown in the following listing is `DeleteScheduledMessagesResponse.xsd` which can be found online at <http://xml.m4u.com.au/2009/DeleteScheduledMessagesResponse.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="ResponseTypes.xsd"/>

  <xsd:element name="deleteScheduledMessagesResponse"
    type="DeleteScheduledMessagesResponseType"/>
</xsd:schema>
```

Listing A.14.1: Delete Scheduled Messages Request XML Schema

A.15 BLOCK NUMBERS REQUEST SCHEMA

Shown in the following listing is BlockNumbersRequest.xsd which can be found online at <http://xml.m4u.com.au/2009/BlockNumbersRequest.xsd>

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="RequestTypes.xsd"/>

  <xsd:element name="blockNumbers" type="BlockNumbersRequestType"/>

</xsd:schema>
```

Listing A.15.1: Block Numbers Request XML Schema

A.16 BLOCK NUMBERS RESPONSE SCHEMA

Shown in the following listing is BlockNumbersResponse.xsd which can be found online at <http://xml.m4u.com.au/2009/BlockNumbersResponse.xsd>

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="ResponseTypes.xsd"/>

  <xsd:element name="blockNumbersResponse"
  type="BlockNumbersResponseType"/>

</xsd:schema>
```

Listing A.16.1: Block Numbers Response XML Schema

A.17 UNBLOCK NUMBERS REQUEST SCHEMA

Shown in the following listing is UnblockNumbersRequest.xsd which can be found online at <http://xml.m4u.com.au/2009/UnblockNumbersRequest.xsd>

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="RequestTypes.xsd"/>

  <xsd:element name="unblockNumbers" type="UnblockNumbersRequestType"/>

</xsd:schema>
```

Listing A.17.1: Unblock Numbers Request XML Schema

A.18 UNBLOCK NUMBERS RESPONSE SCHEMA

Shown in the following listing is UnblockNumbersResponse.xsd which can be found online at <http://xml.m4u.com.au/2009/UnblockNumbersResponse.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="ResponseTypes.xsd"/>

  <xsd:element name="unblockNumbersResponse"
    type="UnblockNumbersResponseType"/>

</xsd:schema>
```

Listing A.18.1: Unblock Numbers Response XML Schema

A.19 GET BLOCKED NUMBERS REQUEST SCHEMA

Shown in the following listing is GetBlockedNumbersRequest.xsd which can be found online at <http://xml.m4u.com.au/2009/GetBlockedNumbersRequest.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="RequestTypes.xsd"/>

  <xsd:element name="getBlockedNumbers"
    type="GetBlockedNumbersRequestType"/>

</xsd:schema>
```

Listing A.19.1: Get Blocked Numbers Request XML Schema

A.20 GET BLOCKED NUMBERS RESPONSE SCHEMA

Shown in the following listing is GetBlockedNumbersResponse.xsd which can be found online at <http://xml.m4u.com.au/2009/GetBlockedNumbersResponse.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="ResponseTypes.xsd"/>

  <xsd:element name="getBlockedNumbersResponse"
    type="GetBlockedNumbersResponseType"/>

</xsd:schema>
```

Listing A.20.1: Get Blocked Numbers Response XML Schema

A.21 FAULT RESPONSE SCHEMA

Shown in the following listing is FaultResponse.xsd which can be found online at <http://xml.m4u.com.au/2009/FaultResponse.xsd>.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="ResponseTypes.xsd"/>

  <xsd:element name="faultResponse" type="FaultResponseType"/>

</xsd:schema>
```

Listing A.13.1: Fault Response XML Schema

A.22 REQUEST TYPES SCHEMA

Shown in the following listing is RequestTypes.xsd which can be found online at <http://xml.m4u.com.au/2009/RequestTypes.xsd>. This schema file is included by all request schemas.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="CommonTypes.xsd"/>

  <!-- Request Types:
       The following types represent the different types of requests
       accepted by the XML Interface.
  -->

  <xsd:complexType name="SendMessageRequestType">
    <xsd:sequence>
      <xsd:element name="authentication" type="AuthenticationType"/>
      <xsd:element name="requestBody" type="SendMessageBodyType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="CheckUserRequestType">
    <xsd:sequence>
      <xsd:element name="authentication" type="AuthenticationType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="CheckRepliesRequestType">
    <xsd:sequence>
      <xsd:element name="authentication" type="AuthenticationType"/>
      <xsd:element name="requestBody" type="CheckRepliesBodyType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="CheckReportsRequestType">
    <xsd:sequence>
      <xsd:element name="authentication" type="AuthenticationType"/>
      <xsd:element name="requestBody" type="CheckReportsBodyType"/>
    </xsd:sequence>
  </xsd:complexType>
```

```

    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="ConfirmRepliesRequestType">
    <xsd:sequence>
      <xsd:element name="authentication" type="AuthenticationType"/>
      <xsd:element name="requestBody" type="ConfirmRepliesBodyType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="ConfirmReportsRequestType">
    <xsd:sequence>
      <xsd:element name="authentication" type="AuthenticationType"/>
      <xsd:element name="requestBody" type="ConfirmReportsBodyType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="DeleteScheduledMessagesRequestType">
    <xsd:sequence>
      <xsd:element name="authentication" type="AuthenticationType"/>
      <xsd:element name="requestBody"
type="DeleteScheduledMessagesBodyType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="BlockNumbersRequestType">
    <xsd:sequence>
      <xsd:element name="authentication" type="AuthenticationType"/>
      <xsd:element name="requestBody" type="BlockNumbersBodyType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="UnblockNumbersRequestType">
    <xsd:sequence>
      <xsd:element name="authentication" type="AuthenticationType"/>
      <xsd:element name="requestBody" type="UnblockNumbersBodyType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="GetBlockedNumbersRequestType">
    <xsd:sequence>
      <xsd:element name="authentication" type="AuthenticationType"/>
      <xsd:element name="requestBody" type="GetBlockedNumbersBodyType"/>
    </xsd:sequence>
  </xsd:complexType>

  <!-- Types used by the Request Types:
    The following types define the types of the elements that
    the Request Types are composed of.
  -->

  <xsd:complexType name="AuthenticationType">
    <xsd:sequence>
      <xsd:element name="userId" type="xsd:string"/>
      <xsd:element name="password" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="SendMessageBodyType">
    <xsd:sequence>
      <xsd:element name="messages" type="MessageListType"/>
    </xsd:sequence>
  </xsd:complexType>

```

```

</xsd:complexType>

<xsd:complexType name="CheckRepliesBodyType">
  <xsd:sequence>
    <xsd:element name="maximumReplies" type="xsd:unsignedInt"
minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CheckReportsBodyType">
  <xsd:sequence>
    <xsd:element name="maximumReports" type="xsd:unsignedInt"
minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ConfirmRepliesBodyType">
  <xsd:sequence>
    <xsd:element name="replies" type="ConfirmReplyListType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ConfirmReportsBodyType">
  <xsd:sequence>
    <xsd:element name="reports" type="ConfirmReportListType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MessageListType">
  <xsd:sequence>
    <xsd:element name="message" type="MessageType"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="sendMode" type="MessageSendModeType"
use="optional" default="normal"/>
</xsd:complexType>

<xsd:simpleType name="MessageSendModeType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="dropAll"/>
    <xsd:enumeration value="dropAllWithErrors"/>
    <xsd:enumeration value="dropAllWithSuccess"/>
    <xsd:enumeration value="normal"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="CheckRepliesType">
  <xsd:sequence>
    <xsd:element name="maximumReplies" type="xsd:unsignedInt"
minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CheckReportListType">
  <xsd:sequence>
    <xsd:element name="maximumReports" type="xsd:unsignedInt"
minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ConfirmReplyListType">
  <xsd:sequence>

```

```

    <xsd:element name="reply" type="ConfirmItemType"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ConfirmReportListType">
  <xsd:sequence>
    <xsd:element name="report" type="ConfirmItemType"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DeleteScheduledMessagesBodyType">
  <xsd:sequence>
    <xsd:element name="messages" type="MessageIdListType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MessageIdListType">
  <xsd:sequence>
    <xsd:element name="message" type="MessageIdType"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MessageIdType">
  <xsd:sequence/>
  <xsd:attribute name="messageId" type="xsd:unsignedInt"
use="required"/>
</xsd:complexType>

<xsd:complexType name="MessageTagType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="name" type="xsd:string" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="MessageTagListType">
  <xsd:sequence>
    <xsd:element name="tag" type="MessageTagType"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MessageType">
  <xsd:all>
    <xsd:element name="origin" type="xsd:string" minOccurs="0"/>
    <xsd:element name="recipients" type="RecipientsType"/>
    <xsd:element name="content" type="xsd:string"/>
    <xsd:element name="scheduled" type="xsd:dateTime" minOccurs="0"/>
    <xsd:element name="deliveryReport" type="xsd:boolean" minOccurs="0"
default="false"/>
    <xsd:element name="validityPeriod" type="xsd:unsignedByte"
minOccurs="0" default="169"/>
    <xsd:element name="tags" type="MessageTagListType" minOccurs="0"/>
  </xsd:all>
  <xsd:attribute name="format" type="MessageFormatType" use="optional"
default="SMS"/>
  <xsd:attribute name="sequenceNumber" type="xsd:unsignedInt"
use="optional" default="0"/>

```

```

    <xsd:anyAttribute processContents="lax"/>
  </xsd:complexType>

  <xsd:complexType name="ConfirmItemType">
    <xsd:sequence/>
    <xsd:attribute name="receiptId" type="xsd:unsignedInt"
use="required"/>
  </xsd:complexType>

  <xsd:complexType name="BlockNumbersBodyType">
    <xsd:sequence>
      <xsd:element name="recipients" type="RecipientsType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="UnblockNumbersBodyType">
    <xsd:sequence>
      <xsd:element name="recipients" type="RecipientsType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="GetBlockedNumbersBodyType">
    <xsd:sequence>
      <xsd:element name="maximumRecipients" type="xsd:unsignedInt"
minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>

```

Listing A.14.1: Request Types XML Schema

A.23 RESPONSE TYPES SCHEMA

Shown in the following listing is ResponseTypes.xsd which can be found online at <http://xml.m4u.com.au/2009/ResponseTypes.xsd>. This schema file is included by all response schemas.

```

<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:include schemaLocation="CommonTypes.xsd"/>

  <!-- Response Types:
    The following types represent the different types of responses
    generated by the XML Interface.
  -->

  <xsd:complexType name="SendMessageResponseType">
    <xsd:sequence>
      <xsd:element name="result" type="SendMessageResultType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="CheckUserResponseType">
    <xsd:sequence>
      <xsd:element name="result" type="CheckUserResultType"/>
    </xsd:sequence>
  </xsd:complexType>

```

```

<xsd:complexType name="CheckRepliesResponseType">
  <xsd:sequence>
    <xsd:element name="result" type="CheckRepliesResultType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CheckReportsResponseType">
  <xsd:sequence>
    <xsd:element name="result" type="CheckReportsResultType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ConfirmRepliesResponseType">
  <xsd:sequence>
    <xsd:element name="result" type="ConfirmRepliesResultType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ConfirmReportsResponseType">
  <xsd:sequence>
    <xsd:element name="result" type="ConfirmReportsResultType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FaultResponseType">
  <xsd:sequence>
    <xsd:element name="error" type="FaultResultType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DeleteScheduledMessagesResponseType">
  <xsd:sequence>
    <xsd:element name="result"
type="DeleteScheduledMessagesResultType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="BlockNumbersResponseType">
  <xsd:sequence>
    <xsd:element name="result" type="BlockNumbersResultType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="UnblockNumbersResponseType">
  <xsd:sequence>
    <xsd:element name="result" type="UnblockNumbersResultType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="GetBlockedNumbersResponseType">
  <xsd:sequence>
    <xsd:element name="result" type="GetBlockedNumbersResultType"/>
  </xsd:sequence>
</xsd:complexType>

<!-- Types used by the Response Types:
  The following types define the types of the elements that
  the Response Types are composed of.
-->

<xsd:complexType name="SendMessageResultType">
  <xsd:sequence>

```

```

        <xsd:element name="accountDetails" type="AccountDetailsType"/>
        <xsd:element name="errors" type="MessageErrorListType"
minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="sent" type="xsd:unsignedInt" use="required"/>
    <xsd:attribute name="scheduled" type="xsd:unsignedInt"
use="required"/>
    <xsd:attribute name="failed" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>

<xsd:complexType name="CheckUserResultType">
    <xsd:sequence>
        <xsd:element name="accountDetails" type="AccountDetailsType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CheckRepliesResultType">
    <xsd:sequence>
        <xsd:element name="replies" type="ReplyListType"/>
    </xsd:sequence>
    <xsd:attribute name="returned" type="xsd:unsignedInt" use="required"/>
    <xsd:attribute name="remaining" type="xsd:unsignedInt"
use="required"/>
</xsd:complexType>

<xsd:complexType name="CheckReportsResultType">
    <xsd:sequence>
        <xsd:element name="reports" type="ReportListType"/>
    </xsd:sequence>
    <xsd:attribute name="returned" type="xsd:unsignedInt" use="required"/>
    <xsd:attribute name="remaining" type="xsd:unsignedInt"
use="required"/>
</xsd:complexType>

<xsd:complexType name="ConfirmRepliesResultType">
    <xsd:attribute name="confirmed" type="xsd:unsignedInt"
use="required"/>
</xsd:complexType>

<xsd:complexType name="ConfirmReportsResultType">
    <xsd:attribute name="confirmed" type="xsd:unsignedInt"
use="required"/>
</xsd:complexType>

<xsd:complexType name="FaultResultType">
    <xsd:sequence/>
    <xsd:attribute name="code" type="FaultErrorCodeType" use="required"/>
</xsd:complexType>

<xsd:complexType name="DeleteScheduledMessagesResultType">
    <xsd:attribute name="unscheduled" type="xsd:unsignedInt"
use="required"/>
</xsd:complexType>

<xsd:simpleType name="FaultErrorCodeType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="authenticationFailed"/>
        <xsd:enumeration value="invalidDataFormat"/>
        <xsd:enumeration value="perDayMessageLimit"/>
        <xsd:enumeration value="requestSizeLimit"/>
    </xsd:restriction>
</xsd:simpleType>

```

```

    <xsd:complexType name="AccountDetailsType">
      <xsd:attribute name="type" type="xsd:string" use="required"
fixed="daily"/>
      <xsd:attribute name="creditLimit" type="xsd:unsignedInt"
use="required"/>
      <xsd:attribute name="creditRemaining" type="xsd:unsignedInt"
use="required"/>
    </xsd:complexType>

    <xsd:complexType name="MessageErrorListType">
      <xsd:sequence>
        <xsd:element name="error" type="MessageErrorType"
maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="MessageErrorType">
      <xsd:sequence>
        <xsd:element name="recipients" type="RecipientsType"/>
        <xsd:element name="content" type="xsd:string" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="code" type="MessageErrorCodeType"
use="required"/>
      <xsd:attribute name="sequenceNumber" type="xsd:unsignedInt"
use="required"/>
    </xsd:complexType>

    <xsd:simpleType name="MessageErrorCodeType">
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="invalidRecipient"/>
        <xsd:enumeration value="recipientBlocked"/>
        <xsd:enumeration value="emptyMessageContent"/>
        <xsd:enumeration value="messageLengthLimit"/>
        <xsd:enumeration value="other"/>
      </xsd:restriction>
    </xsd:simpleType>

    <xsd:complexType name="ReplyListType">
      <xsd:sequence>
        <xsd:element name="reply" type="ReplyType" minOccurs="0"
maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="ReplyType">
      <xsd:sequence>
        <xsd:element name="origin" type="xsd:string"/>
        <xsd:element name="received" type="xsd:dateTime"/>
        <xsd:element name="content" type="xsd:string"/>
      </xsd:sequence>
      <xsd:attribute name="format" type="MessageFormatType" use="required"/>
      <xsd:attribute name="uid" type="xsd:unsignedInt" use="required"/>
      <xsd:attribute name="receiptId" type="xsd:unsignedInt"
use="required"/>
    </xsd:complexType>

    <xsd:complexType name="ReportListType">
      <xsd:sequence>
        <xsd:element name="report" type="ReportType" minOccurs="0"
maxOccurs="unbounded"/>
      </xsd:sequence>

```

```

</xsd:complexType>

<xsd:complexType name="ReportType">
  <xsd:sequence>
    <xsd:element name="recipient" type="xsd:string"/>
    <xsd:element name="timestamp" type="xsd:dateTime"/>
  </xsd:sequence>
  <xsd:attribute name="uid" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="receiptId" type="xsd:unsignedInt"
use="required"/>
  <xsd:attribute name="status" type="DeliveryStatusType"
use="required"/>
</xsd:complexType>

<xsd:simpleType name="DeliveryStatusType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="delivered"/>
    <xsd:enumeration value="pending"/>
    <xsd:enumeration value="failed"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="BlockNumbersResultType">
  <xsd:sequence>
    <xsd:element name="errors" type="MessageErrorListType"
minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="blocked" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="failed" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>

<xsd:complexType name="UnblockNumbersResultType">
  <xsd:sequence>
    <xsd:element name="errors" type="MessageErrorListType"
minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="unblocked" type="xsd:unsignedInt"
use="required"/>
  <xsd:attribute name="failed" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>

<xsd:complexType name="GetBlockedNumbersResultType">
  <xsd:sequence>
    <xsd:element name="recipients" type="RecipientsType"/>
  </xsd:sequence>
  <xsd:attribute name="found" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="returned" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>

</xsd:schema>

```

Listing A.15.1: Response Types XML Schema

A.24 COMMON TYPES SCHEMA

Shown in the following listing is CommonTypes.xsd which can be found online at <http://xml.m4u.com.au/2009/CommonTypes.xsd>. This schema file is included by both RequestTypes.xsd and ResponseTypes.xsd.

```

<?xml version="1.0" encoding="utf-8"?>
<xsd:schema

```

```

attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:complexType name="RecipientsType">
    <xsd:sequence>
      <xsd:element name="recipient" type="RecipientType"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="RecipientType">
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="uid" type="xsd:unsignedInt" use="optional"
default="0"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>

  <xsd:simpleType name="MessageFormatType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="SMS"/>
      <xsd:enumeration value="voice"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>

```

Listing A.16.1: Common Types XML Schema

Appendix B: WSDL File

This appendix provides a listing of the MessageMedia Messaging Web Service WSDL file.

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions
  targetNamespace="http://xml.m4u.com.au/2009"
  xmlns="http://xml.m4u.com.au/2009"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:api="http://xml.m4u.com.au/2009"
  xmlns:ws-i="http://ws-i.org/schemas/conformanceClaim/"
  >

  <wsdl:types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/SendMessages.xsd"/>
      <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/CheckUser.xsd"/>
      <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/CheckReplies.xsd"/>
      <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/CheckReports.xsd"/>
      <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/ConfirmReplies.xsd"/>
      <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/ConfirmReports.xsd"/>
      <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/BlockNumbers.xsd"/>
    </xsd:schema>
  </wsdl:types>

```

```

    <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/UnblockNumbers.xsd"/>
    <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/GetBlockedNumbers.xsd"/>
    <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/SendMessagesResponse.xsd"/>
    <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/CheckUserResponse.xsd"/>
    <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/CheckRepliesResponse.xsd"/>
    <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/CheckReportsResponse.xsd"/>
    <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/ConfirmRepliesResponse.xsd"/>
    <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/ConfirmReportsResponse.xsd"/>
    <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/BlockNumbersResponse.xsd"/>
    <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/UnblockNumbersResponse.xsd"/>
    <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/GetBlockedNumbersResponse.xsd"/>
    <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/FaultResponse.xsd"/>
    <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/DeleteScheduledMessages.xsd"/>
    <xsd:import namespace="http://xml.m4u.com.au/2009"
schemaLocation="http://xml.m4u.com.au/2009/DeleteScheduledMessagesResponse.x
sd"/>
  </xsd:schema>
</wsdl:types>

<wsdl:message name="sendMessagesRequest">
  <wsdl:part name="parameters" element="api:sendMessages"/>
</wsdl:message>
<wsdl:message name="sendMessagesResponse">
  <wsdl:part name="parameters" element="api:sendMessagesResponse"/>
</wsdl:message>

<wsdl:message name="checkUserRequest">
  <wsdl:part name="parameters" element="api:checkUser"/>
</wsdl:message>
<wsdl:message name="checkUserResponse">
  <wsdl:part name="parameters" element="api:checkUserResponse"/>
</wsdl:message>

<wsdl:message name="checkRepliesRequest">
  <wsdl:part name="parameters" element="api:checkReplies"/>
</wsdl:message>
<wsdl:message name="checkRepliesResponse">
  <wsdl:part name="parameters" element="api:checkRepliesResponse"/>
</wsdl:message>

<wsdl:message name="checkReportsRequest">
  <wsdl:part name="parameters" element="api:checkReports"/>
</wsdl:message>
<wsdl:message name="checkReportsResponse">
  <wsdl:part name="parameters" element="api:checkReportsResponse"/>
</wsdl:message>

<wsdl:message name="confirmRepliesRequest">
  <wsdl:part name="parameters" element="api:confirmReplies"/>

```

```

</wsdl:message>
<wsdl:message name="confirmRepliesResponse">
  <wsdl:part name="parameters" element="api:confirmRepliesResponse"/>
</wsdl:message>

<wsdl:message name="confirmReportsRequest">
  <wsdl:part name="parameters" element="api:confirmReports"/>
</wsdl:message>
<wsdl:message name="confirmReportsResponse">
  <wsdl:part name="parameters" element="api:confirmReportsResponse"/>
</wsdl:message>

<wsdl:message name="faultResponse">
  <wsdl:part name="detail" element="api:faultResponse"/>
</wsdl:message>

<wsdl:message name="deleteScheduledMessagesRequest">
  <wsdl:part name="parameters" element="api:deleteScheduledMessages"/>
</wsdl:message>
<wsdl:message name="deleteScheduledMessagesResponse">
  <wsdl:part name="parameters"
element="api:deleteScheduledMessagesResponse"/>
</wsdl:message>

<wsdl:message name="blockNumbersRequest">
  <wsdl:part name="parameters" element="api:blockNumbers"/>
</wsdl:message>
<wsdl:message name="blockNumbersResponse">
  <wsdl:part name="parameters" element="api:blockNumbersResponse"/>
</wsdl:message>

<wsdl:message name="unlockNumbersRequest">
  <wsdl:part name="parameters" element="api:unlockNumbers"/>
</wsdl:message>
<wsdl:message name="unlockNumbersResponse">
  <wsdl:part name="parameters" element="api:unlockNumbersResponse"/>
</wsdl:message>

<wsdl:message name="getBlockedNumbersRequest">
  <wsdl:part name="parameters" element="api:getBlockedNumbers"/>
</wsdl:message>
<wsdl:message name="getBlockedNumbersResponse">
  <wsdl:part name="parameters" element="api:getBlockedNumbersResponse"/>
</wsdl:message>

<wsdl:portType name="MessageMediaServiceInterface">
  <wsdl:operation name="sendMessages">
    <wsdl:input name="sendMessagesRequest"
message="tsendMessageRequest"/>
    <wsdl:output name="sendMessagesResponse"
message="tsendMessageResponse"/>
    <wsdl:fault name="sendMessagesFault" message="tfaultResponse"/>
  </wsdl:operation>
  <wsdl:operation name="checkUser">
    <wsdl:input name="checkUserRequest" message="tcheckUserRequest"/>
    <wsdl:output name="checkUserResponse" message="tcheckUserResponse"/>
    <wsdl:fault name="checkUserFault" message="tfaultResponse"/>
  </wsdl:operation>
  <wsdl:operation name="checkReplies">
    <wsdl:input name="checkRepliesRequest"
message="tcheckRepliesRequest"/>
    <wsdl:output name="checkRepliesResponse"

```

```

message="tcheckRepliesResponse"/>
  <wsdl:fault name="checkRepliesFault" message="tfaultResponse"/>
</wsdl:operation>
  <wsdl:operation name="checkReports">
    <wsdl:input name="checkReportsRequest"
message="tcheckReportsRequest"/>
    <wsdl:output name="checkReportsResponse"
message="tcheckReportsResponse"/>
    <wsdl:fault name="checkReportsFault" message="tfaultResponse"/>
  </wsdl:operation>
  <wsdl:operation name="confirmReplies">
    <wsdl:input name="confirmRepliesRequest"
message="tconfirmRepliesRequest"/>
    <wsdl:output name="confirmRepliesResponse"
message="tconfirmRepliesResponse"/>
    <wsdl:fault name="confirmRepliesFault" message="tfaultResponse"/>
  </wsdl:operation>
  <wsdl:operation name="confirmReports">
    <wsdl:input name="confirmReportsRequest"
message="tconfirmReportsRequest"/>
    <wsdl:output name="confirmReportsResponse"
message="tconfirmReportsResponse"/>
    <wsdl:fault name="confirmReportsFault" message="tfaultResponse"/>
  </wsdl:operation>
  <wsdl:operation name="deleteScheduledMessages">
    <wsdl:input name="deleteScheduledMessagesRequest"
message="tdeleteScheduledMessagesRequest"/>
    <wsdl:output name="deleteScheduledMessagesResponse"
message="tdeleteScheduledMessagesResponse"/>
    <wsdl:fault name="deleteScheduledMessagesFault"
message="tfaultResponse"/>
  </wsdl:operation>
  <wsdl:operation name="blockNumbers">
    <wsdl:input name="blockNumbersRequest"
message="tblockNumbersRequest"/>
    <wsdl:output name="blockNumbersResponse"
message="tblockNumbersResponse"/>
    <wsdl:fault name="blockNumbersFault" message="tfaultResponse"/>
  </wsdl:operation>
  <wsdl:operation name="unblockNumbers">
    <wsdl:input name="unblockNumbersRequest"
message="tunblockNumbersRequest"/>
    <wsdl:output name="unblockNumbersResponse"
message="tunblockNumbersResponse"/>
    <wsdl:fault name="unblockNumbersFault" message="tfaultResponse"/>
  </wsdl:operation>
  <wsdl:operation name="getBlockedNumbers">
    <wsdl:input name="getBlockedNumbersRequest"
message="tgetBlockedNumbersRequest"/>
    <wsdl:output name="getBlockedNumbersResponse"
message="tgetBlockedNumbersResponse"/>
    <wsdl:fault name="getBlockedNumbersFault" message="tfaultResponse"/>
  </wsdl:operation>
</wsdl:portType>

  <wsdl:binding name="MessageMediaServiceBinding"
type="tMessageMediaServiceInterface">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="sendMessages">
      <soap:operation soapAction="http://xml.m4u.com.au/2009/sendMessages"/>
      <wsdl:input name="sendMessagesRequest">

```

```

    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="sendMessagesResponse">
    <soap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="sendMessagesFault">
    <soap:fault name="sendMessagesFault" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="checkUser">
  <soap:operation soapAction="http://xml.m4u.com.au/2009/checkUser"/>
  <wsdl:input name="checkUserRequest">
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="checkUserResponse">
    <soap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="checkUserFault">
    <soap:fault name="checkUserFault" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="checkReplies">
  <soap:operation soapAction="http://xml.m4u.com.au/2009/checkReplies"/>
  <wsdl:input name="checkRepliesRequest">
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="checkRepliesResponse">
    <soap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="checkRepliesFault">
    <soap:fault name="checkRepliesFault" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="checkReports">
  <soap:operation soapAction="http://xml.m4u.com.au/2009/checkReports"/>
  <wsdl:input name="checkReportsRequest">
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="checkReportsResponse">
    <soap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="checkReportsFault">
    <soap:fault name="checkReportsFault" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="confirmReplies">
  <soap:operation
soapAction="http://xml.m4u.com.au/2009/confirmReplies"/>
  <wsdl:input name="confirmRepliesRequest">
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="confirmRepliesResponse">
    <soap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="confirmRepliesFault">
    <soap:fault name="confirmRepliesFault" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="confirmReports">
  <soap:operation
soapAction="http://xml.m4u.com.au/2009/confirmReports"/>
  <wsdl:input name="confirmReportsRequest">

```

```

        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="confirmReportsResponse">
        <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="confirmReportsFault">
        <soap:fault name="confirmReportsFault" use="literal"/>
    </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="deleteScheduledMessages">
    <soap:operation
soapAction="http://xml.m4u.com.au/2009/soap/deleteScheduledMessages"/>
    <wsdl:input name="deleteScheduledMessagesRequest">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="deleteScheduledMessagesResponse">
        <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="deleteScheduledMessagesFault">
        <soap:fault name="deleteScheduledMessagesFault" use="literal"/>
    </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="blockNumbers">
    <soap:operation soapAction="http://xml.m4u.com.au/2009/blockNumbers"/>
    <wsdl:input name="blockNumbersRequest">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="blockNumbersResponse">
        <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="blockNumbersFault">
        <soap:fault name="blockNumbersFault" use="literal"/>
    </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="unblockNumbers">
    <soap:operation
soapAction="http://xml.m4u.com.au/2009/unblockNumbers"/>
    <wsdl:input name="unblockNumbersRequest">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="unblockNumbersResponse">
        <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="unblockNumbersFault">
        <soap:fault name="unblockNumbersFault" use="literal"/>
    </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="getBlockedNumbers">
    <soap:operation
soapAction="http://xml.m4u.com.au/2009/getBlockedNumbers"/>
    <wsdl:input name="getBlockedNumbersRequest">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="getBlockedNumbersResponse">
        <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="getBlockedNumbersFault">
        <soap:fault name="getBlockedNumbersFault" use="literal"/>
    </wsdl:fault>
</wsdl:operation>
</wsdl:binding>

```

```
<wsdl:service name="MessageMediaService">
  <wsdl:port name="MessageMediaServiceEndPoint"
binding="tMessageMediaServiceBinding">
  <wsdl:documentation>
    <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.0"/>
  </wsdl:documentation>
  <soap:address location="http://soap.m4u.com.au"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

Listing B.1.1: MessageMedia Messaging Web Service WSDL File

Appendix C: Redundant Service

Note that while the MessageMedia domains will automatically fail-over to a redundant backup site in the case of an outage, due to limitations of the DNS technology itself this can take up to 70 seconds. We recommend that any customers relying on immediate delivery integrate logic to switch to the backup site immediately if the primary domain is not responding. The backup site will only accept requests when the primary is off-line, so it may take up to 10 seconds for it to become available. For critical applications, please contact MessageMedia Support to have your implementation white-listed so the backup server is always available.

Primary domains: <https://soap.m4u.com.au> & <https://xml.m4u.com.au>

Backup domain: <https://soap-backup.m4u.com.au> & <https://xml-backup.m4u.com.au>

Contact us

Australia

Level 22
385 Bourke Street
Melbourne VIC 3000

UK

Level 6
52 Grosvenor Gardens
Victoria London SW1W 0AU

USA

Level Five
One Embarcadero Center
San Francisco, CA 94111

Sales: 1800 009 767
Support: 1800 155 228

Fax: 03 8612 3689

Sales: 0808 234 8246
Support: 0808 234 4874

Fax: 0208 082 5091

Sales: 888-799-9767
Support: 866-751-8337

Fax: 646 862 8887

International

+61 3 9600 0777

Email: support@message-media.com