



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

messagingPeople Developer APIs for WSDL (SOAP Web Services)

Version 2.11 (Dec 2011)

Copyright © messagingPeople 2006-2012. All documentation provided remains the property of messagingPeople. You cannot copy, reproduce or distribute this document without a written authorisation from messagingPeople.

For support related developer questions contact: support@messagingpeople.com or visit the Frequently Asked Questions list (FAQ) and support forum at <http://www.messagingpeople.com/faq/index.php>.



PO BOX 809
 Glen Waverley
 VIC 3150
 AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

Table of Contents

1. GENERAL.....	5
1.1 SUPPORTED ENVIRONMENTS.....	5
1.2 SECURE WEB SERVICES SUPPORT	5
1.3 SAMPLE SOURCE CODE	5
1.4 GETTING STARTED.....	5
1.5 BINARY DATA ENCODING BASE64BINARY	6
2. DEVELOPERS APIS	7
2.1 CREATING A SESSION.....	7
2.2 ENDING A SESSION.....	7
2.3 GET VERSION.....	7
2.4 GET ACCOUNT BALANCE.....	8
2.5 GET ACCOUNT HISTORY	8
2.6 CHANGE ACCOUNT PASSWORD.....	9
2.7 GET MESSAGINGPEOPLE NEWS	9
2.8 GET EMAIL MESSAGING CONFIGURATION SETTINGS.....	9
2.9 EDIT EMAIL MESSAGING CONFIGURATION SETTINGS.....	10
2.10 START SCHEDULING MESSAGES	11
2.11 STOP SCHEDULING MESSAGES.....	11
2.12 SEND SMS.....	12
<i>Plain text SMS:</i>	13
<i>Unicode SMS:</i>	13
2.13 SEND MMS.....	14
2.14 SEND WAP	16
2.15 SEND FAX.....	17
2.16 GET MESSAGE STATUS	18
2.17 SET SENDER ID.....	18
2.18 ADD NEW USER (PARTNER ENABLED ACCOUNTS).....	19
2.19 EDIT SUB USER (PARTNER ENABLED ACCOUNTS).....	19
2.20 ADD NEW EVENT (PARTNER ENABLED ACCOUNTS).....	19
2.21 GET EMAIL MESSAGING CONFIGURATION SETTINGS (PARTNER ENABLED ACCOUNTS).....	20
2.22 EDIT EMAIL MESSAGING CONFIGURATION SETTINGS (PARTNER ENABLED ACCOUNTS).....	21
2.23 GET COST BY COUNTRY (PARTNER ENABLED ACCOUNTS).....	22
2.24 GET COST BY PHONE NUMBER (PARTNER ENABLED ACCOUNTS).....	24
2.25 CREATING A SUB USER SESSION (PARTNER ENABLED ACCOUNTS).....	24
2.26 ENDING A SUB USER SESSION (PARTNER ENABLED ACCOUNTS).....	25
2.27 PARTNERSHIP API.....	26



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

2.28	QR-CODES (QUICK RESPONSE CODES) API	26
3.	APPENDICES	27
3.1	ERROR CODES	27
3.2	CURRENCY CODES	27
3.3	PACKAGE TYPES	28



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

History

Date	Description
01 Aug 2006	Initial release.
27 Sep 2006	Updated the document logo.
16 May 2008	Added partner and new client APIs.
08 Jun 2008	Added Secure WSDL documentation.
09 Aug 2008	Added partnership and QR-Codes API information.
22 Dec 2008	Added long SMS and Unicode SMS details.
07 Jan 2009	Removed confusion surrounding "Email Code" terminology. This has now been renamed to "Identity Code".
15 May 2010	Updated documentation templates and new method setSenderId



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

1. General

1.1 Supported Environments

By connecting to the messagingPeople gateway via WSDL (SOAP Web Services) Interface means that all operating systems can use the interfaces published. This is the preferred technology to use for interfacing with the messagingPeople Gateway.

1.2 Secure Web Services Support

We support web services over HTTP or HTTPS protocol. The APIs between the two are exactly the same – just the endpoints to connect to the WSDL server are different.

If using the secure WSDL then you must ensure that you perform all the SSL handshaking to set-up a secure transport prior to invoking the WSDL API methods. Most libraries connecting to WSDL interfaces automatically do this upon detecting the HTTPS endpoint.

1.3 Sample Source Code

The sample source code can be found by downloading the messagingPeople SDK. The messagingPeople SDK provides all the online documentation, sample projects for ASP, C#.NET, c-shell scripts, HTML, JAVA, JSP, perl, PHP, VB, VB.NET, VC++, C++.NET and other development languages for sending SMS, WAP, MMS and faxes.

Download the SDK package from:

<http://www.messagingpeople.com/download/messagingPeopleSDK/setup.exe>.

1.4 Getting Started

The WSDL details for connecting to the messagingPeople gateway are published at <http://www.messagingpeople.com/wsd/messagingPeopleGateway.wsdl> or http://www.messagingpeople.com/wsd/messagingPeopleGateway_Secure.wsdl if you wish to use secure web services. Simply create a new .NET or Java project and point to the published messagingPeople Gateway WSDL.

Code wrappers for connecting to the messagingPeople Gateway will be automatically generated by the application IDE.

Consequently, there are also many WSDL parsers that generate source code for interacting with WSDL servers for inclusion into your own development projects. For example: gSOAP (See <http://gsoap2.sourceforge.net/>) generates cross platform source code in C++ for connecting to a WSDL Servers.



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

Useful tools for seeing the published WSDL services can be found at:
<http://www.mgateway.com/php/mgw/wsdlvalidator.php>.

1.5 Binary Data Encoding base64Binary

When sending binary data to the WSDL server you must first encode the content into a base64Binary blob. Simply set the size of the base64Binary to be the length of the data your transmitting and the data to point to the data blob to transmit.



messagingPeople

2. Developers APIs

2.1 Creating a Session

All communication to the messagingPeople gateway requires you to login and create a session. Sessions with the gateway are timed and are automatically disconnected after a period of idle activity or until the end session request is sent.

It is important that once you have finished you send the end session request to reclaim resources allocated.

sessionStart		
Parameter 1 <i>sSession</i>	Session	[out] xsd:string
Parameter 2 <i>sUser</i>	Login	[in] xsd:string
Parameter 3 <i>sPassword</i>	Password	[in] xsd:string
Parameter 4 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.2 Ending a Session

Once interaction with the messagingPeople gateway is completed. Developers should end session to reclaim resources allocated.

sessionEnd		
Parameter 1 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 2 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.3 Get Version

Retrieves the version of your messagingPeople gateway COM object.

getVersion



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

Parameter 1 <i>nVersion</i>	Version Number	[out] xsd:integer
--------------------------------	----------------	-------------------

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.4 Get Account Balance

This method returns the account balance outstanding associated with this session.

getAccountBalance		
Parameter 1 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 2 <i>dValue</i>	Balance	[out] xsd:double
Parameter 3 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.5 Get Account History

This method returns the account history for the last 1000 transactions.

getAccountHistory		
Parameter 1 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 2 <i>sXmlAccountHistory</i>	XML encoding of the account history. This will contain ID, Date, Amount (or N/A) and description of the entry.	[out] xsd:string
Parameter 3 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.



messagingPeople

2.6 Change Account Password

This API allows the current account password to be changed. However, this requires the current password to be known (it is not possible to change to a new password without having the original password first).

setAccountPassword		
Parameter 1 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 2 <i>sOldPassword</i>	Old Password	[in] xsd:string
Parameter 3 <i>sNewPassword</i>	New Password	[in] xsd:string
Parameter 4 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.7 Get messagingPeople News

This method returns the latest messagingPeople News. Content returned back is a RTF string (Rich Text Format).

getNews		
Parameter 1 <i>sNews</i>	News	[out] xsd:string
Parameter 2 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.8 Get Email Messaging Configuration Settings

This method returns the current email messaging configuration settings.

getEmailMessaging		
Parameter 1 <i>bEmailEnable</i>	Email messaging enabled flag	[out] xsd:Boolean
Parameter 2 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 3 <i>sEmailCode</i>	Identity Code	[out] xsd:string



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

Parameter 4 <i>bEmailFilter</i>	Perform email filtering flag	[out] xsd:Boolean
Parameter 5 <i>sEmailFilter1</i>	Email filter address 1	[out] xsd:string
Parameter 6 <i>sEmailFilter2</i>	Email filter address 2	[out] xsd:string
Parameter 7 <i>sEmailFilter3</i>	Email filter address 3	[out] xsd:string
Parameter 8 <i>bEmailSmsSplit</i>	Split long emails into multiple SMS flag	[out] xsd:Boolean
Parameter 9 <i>bEmailEndStop</i>	Terminate emails by special stop word ##END## flag	[out] xsd:Boolean
Parameter 10 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer
Parameter 11 <i>bEmailEndDoubleLine</i>	Terminate emails by double black lines flag	[out] xsd:Boolean

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.9 Edit Email Messaging Configuration Settings

This method allows you to configure and change the current email messaging configuration settings. This is useful if you wish to implement your own messaging client software.

editEmailMessaging		
Parameter 1 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 2 <i>sEmailCode</i>	Identity Code	[in] xsd:string
Parameter 3 <i>bEmailFilter</i>	Perform email filtering flag	[in] xsd:Boolean
Parameter 4 <i>sEmailFilter1</i>	Email filter address 1	[in] xsd:string
Parameter 5 <i>sEmailFilter2</i>	Email filter address 2	[in] xsd:string
Parameter 6 <i>sEmailFilter3</i>	Email filter address 3	[in] xsd:string
Parameter 7 <i>bEmailSmsSplit</i>	Split long emails into multiple SMS flag	[in] xsd:Boolean
Parameter 8 <i>bEmailEndStop</i>	Terminate emails by special stop word ##END## flag	[in] xsd:Boolean
Parameter 9	Error Code (See Appendices	[out] xsd:integer



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

<i>nErrorCode</i>	1)	
Parameter 10 <i>bEmailDoubleLine</i>	Terminate emails by double black lines flag	[in] xsd:Boolean
Parameter 11 <i>bEmailEnable</i>	Email messaging enabled flag	[in] xsd:Boolean

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.10 Start Scheduling Messages

Call this method to set the schedule time that you want for messages sent after this call. Scheduling will stop once Stop Schedule Messages method call is called.

scheduleMessageEnable		
Parameter 1 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer
Parameter 2 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 3 <i>sDateTimeGmt</i>	Schedule time in GMT to send messages. This is in the format "YYYY/mm/dd HH:MM:SS" string where YYYY is the year, mm is the month, dd is the day, HH is the hour, MM is the minutes and SS is the seconds. HH range is [0, 24).	[in] xsd:string

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.11 Stop Scheduling Messages

Stops scheduling messages enabled via the Start Scheduling Messages.

scheduleMessageDisable		
Parameter 1 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 2 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.12 Send SMS

This method sends SMS to the list of international phones numbers provided. Phone numbers must be written in international format and separated by a semi-colon.

Additionally this method supports long SMS. What this means is that long messages will be broken into several SMS messages and then concatenated automatically on the receiver's mobile phone as one single message (most newer generation mobile phones already support this feature).

You can send plain text SMS or Unicode SMS. Differences are as follows:



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

Plain text SMS:

Plain text SMS has the maximum length of 760 characters (i.e 5 SMS each with a limit of 160 bytes of data for communication). If you send messages longer than 160 characters then this would be classified as a long SMS and your long message will be automatically broken into several SMS and rejoined as a single message on the receiver's mobile phone.

Unicode SMS:

Unicode messages allow information containing non-English character-set to be transmitted and displayed correctly on the recipient's mobile phone. Languages such as Chinese, Japanese, Korean, Indian, Arabic etc would need to be encoded using Unicode.

Unicode SMS has the maximum length of 335 characters (i.e 5 SMS each with a limit of 160 bytes of data for communication). If you send messages longer than 70 characters then this would be classified as a long SMS and your long message will be automatically broken into several SMS and rejoined as a single message on the receiver's mobile phone.

When sending Unicode messages you need to ensure that your message is in the format as followed:

`%%UNICODE%%<<Hex 1>><<Hex 2>><<Hex 3>> etc`

For example to send the message containing two characters: "你好" which has the Unicode hex codes of each character as 4F60h 597Dh (each character is represented by 2 bytes in hexadecimal – i.e four hexadecimal digits). Thus the final message string to pass to the method would be written as:

`%%UNICODE%%4F60597D`

Do not pass wide character strings or Unicode strings or the likes. Simply just convert each character set into the hexadecimal equivalent.

When sending SMS you can also specify whether this is a flash SMS or a normal SMS message.

Changing the sender SMS number is controlled by logging onto the website portal and specifying the sender SMS number that you want to appear when sending the SMS message. When recipients reply to SMS it will be directed to the sender SMS number (provided that your sender SMS number is valid).

The receipt for each SMS successfully delivered to the messagingPeople gateway will be returned for each phone number otherwise FAILED will be returned.



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

Successful delivery to the gateway does not mean the SMS itself was successfully delivered to the end recipient. To determine the actual message delivery status call the Get Status method and pass the associated receipt.

For example sending a SMS with the list of phone numbers set to:

```
+61413014821; +61413014822; +6241301145
```

Will return in the list of receipts as (in this example +61413014822 failed):

```
+61413014821:1-20050805233244-PSOCIOSLVV, +61413014822:FAILED,  
+6241301145:1-20050811201214-QRJATYDIMO
```

sendSms		
Parameter 1 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 2 <i>sMessage</i>	Message for plain text or Unicode message. See details above.	[in] xsd:string
Parameter 3 <i>bFlash</i>	Flash	[in] Boolean
Parameter 4 <i>sListPhoneNumbers</i>	List of Phone Numbers (must be in international format)	[in] xsd:string
Parameter 5 <i>sListReceipts</i>	List of Receipts	[out] xsd:string
Parameter 6 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.13 Send MMS

This method sends MMS to the list of international phones numbers provided. Phone numbers must be written in international format and separated by a semi-colon.

When sending MMS you can only send files of JPEG, GIF and MIDI types. We do automatically convert other images of unknown types to JPEG formats (see our FAQ online for the list of images automatically converted).

File types are determined by the extension of the file name. Thus your file name should have the appropriate extension given.



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

An optional description can be sent along with your MMS.

Changing the sender MMS number is controlled by logging onto the website portal and specifying the sender MMS number that you want to appear when sending the MMS message. When recipients reply to MMS it will be directed to the sender MMS number (provided that your sender MMS number is valid).

The receipt for each MMS successfully delivered to the messagingPeople gateway will be returned for each phone number otherwise FAILED will be returned.

Successful delivery to the gateway does not mean the MMS itself was successfully delivered to the end recipient. To determine the actual message delivery status call the Get Status method and pass the associated receipt.

For example sending a MMS with the list of phone numbers set to:

```
+61413014821; +61413014822; +6241301145
```

Will return in the list of receipts as (in this example +61413014822 failed):

```
+61413014821:1-20050805233244-PSOCIOSLVV, +61413014822:FAILED,  
+6241301145:1-20050811201214-QRJATYDIMO
```

sendMms		
Parameter 1 <i>sListReciepts</i>	List of Receipts	[out] xsd:string
Parameter 2 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 3 <i>sDescription</i>	Description (Keep this empty for no description)	[in] xsd:string
Parameter 4 <i>sFileName</i>	File Name	[in] xsd:string
Parameter 5 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer
Parameter 6 <i>oData</i>	File Data	[in] xsd:base64Binary
Parameter 7 <i>sListPhoneNumbers</i>	List of Phone Numbers (must be in international format)	[in] xsd:string

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

2.14 Send WAP

This method sends WAP to the list of international phones numbers provided. Phone numbers must be written in international format and separated by a semi-colon.

When sending WAP you can send content of any type. Images are automatically converted to JPEG types (smaller in size) as described on the in the FAQ online. Ring tones, Java Games, videos and other content are simply passed to the destination mobile untouched.

A message should be provided with the WAP message itself.

Changing the sender WAP number is controlled by logging onto the website portal and specifying the sender WAP number that you want to appear when sending the WAP message. When recipients reply to WAP it will be directed to the sender WAP number (provided that your sender WAP number is valid).

The receipt for each WAP successfully delivered to the messagingPeople gateway will be returned for each phone number otherwise FAILED will be returned.

Successful delivery to the gateway does not mean the WAP itself was successfully delivered to the end recipient. To determine the actual message delivery status call the Get Status method and pass the associated receipt.

For example sending a WAP with the list of phone numbers set to:

```
+61413014821; +61413014822; +6241301145
```

Will return in the list of receipts as (in this example +61413014822 failed):

```
+61413014821:1-20050805233244-PSOCIOSLVV, +61413014822:FAILED,  
+6241301145:1-20050811201214-QRJTATYDIMO
```

SendWap		
Parameter 1 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 2 <i>sMessage</i>	Message	[in] xsd:string
Parameter 3 <i>sFileName</i>	File Name	[in] xsd:string
Parameter 4 <i>oData</i>	File Data	[in] xsd:base64Binary
Parameter 5 <i>sListPhoneNumbers</i>	List of Phone Numbers (must be in international format)	[in] xsd:string



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

Parameter 6 <i>sListReciepts</i>	List of Receipts	[out] xsd:string
Parameter 7 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.15 Send FAX

This method sends FAX to the list of international phones numbers provided. Phone numbers must be written in international format and separated by a semi-colon.

When sending FAX you can send content in the supported types as covered in the FAQ online. Unsupported formats will simply be rejected. Documents must be unencrypted with no password protections.

Changing the sender FAX number is controlled by logging onto the website portal and specifying the sender FAX number that you want to appear when sending the FAX message.

The receipt for each FAX successfully delivered to the messagingPeople gateway will be returned for each phone number otherwise FAILED will be returned.

Successful delivery to the gateway does not mean the FAX itself was successfully delivered to the end recipient. To determine the actual message delivery status call the Get Status method and pass the associated receipt.

For example sending a FAX with the list of phone numbers set to:

```
+61413014821; +61413014822; +6241301145
```

Will return in the list of receipts as (in this example +61413014822 failed):

```
+61413014821:1-20050805233244-PSOCIOSLVV, +61413014822:FAILED,  
+6241301145:1-20050811201214-QRJATYDIMO
```

sendFax		
Parameter 1 <i>sListReciepts</i>	List of Receipts	[out] xsd:string
Parameter 2 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 3	File Name	[in] xsd:string



messagingPeople

<i>sFileName</i>		
Parameter 4 <i>oData</i>	File Data	[in] xsd:base64Binary
Parameter 5 <i>sListPhoneNumbers</i>	List of Phone Numbers (must be in international format)	[in] xsd:string
Parameter 6 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.16 Get Message Status

This method gets the Status of the message sent.

getStatus		
Parameter 1 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 2 <i>sReciept</i>	Receipt	[in] xsd:string
Parameter 3 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer
Parameter 4 <i>sStatus</i>	Status of the message	[out] xsd:string

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.17 Set Sender ID

For sending SMS and Direct MMS you can dynamically set your Sender ID to use. Ensure that your account has been enabled to set the Sender ID dynamically otherwise you can change your Sender ID by logging online and setting your Sender ID account preferences.

setSetSenderId		
Parameter 1 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 2 <i>sSenderId</i>	Sender ID	[in] xsd:string
Parameter 3 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.18 Add New User (Partner Enabled Accounts)

This method allows partner account holders to create sub account holders.

partnerAddUser		
Parameter 1 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer
Parameter 2 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 3 <i>sUser</i>	The new user name for sub account	[in] xsd:string
Parameter 4 <i>sEmail</i>	The email of the sub user account holder	[in] xsd:string

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.19 Edit Sub User (Partner Enabled Accounts)

This method allows partner account holders to create sub account holders.

partnerEditUser		
Parameter 1 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer
Parameter 2 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 3 <i>sUser</i>	The updated user name for sub account	[in] xsd:string
Parameter 4 <i>sEmail</i>	The updated email of the sub user account holder	[in] xsd:string

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.20 Add New Event (Partner Enabled Accounts)

This method allows partners to be able to credit / debit a sub account holder or place an informative description text into the sub account holder's history.



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

This allows partners to resell the services and effectively credit their sub account holders so that they can do messaging. Sub account holders can only be enabled for messaging only when they have a positive account balance.

partnerAddEvent		
Parameter 1 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer
Parameter 2 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 3 <i>sUser</i>	The sub account user name	[in] xsd:string
Parameter 4 <i>sEventType</i>	The event type: <i>Debit</i> : Debit account <i>Credit</i> : Credit account <i>Message</i> : Just message only	[in] xsd:string
Parameter 5 <i>sPackageType</i>	The package type to use if crediting the sub account	[in] xsd:string
Parameter 6 <i>dAmount</i>	The amount to use if crediting or debiting the sub account	[in] xsd:double
Parameter 7 <i>sDescription</i>	The description for the entry	[in] xsd:string

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.21 Get Email Messaging Configuration Settings (Partner Enabled Accounts)

This method allows partner account holders to get the current email messaging settings of the sub account holders

partnerGetEmailMessaging		
Parameter 1 <i>bEmailEnable</i>	Email messaging enabled flag	[out] xsd:Boolean
Parameter 2 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 3 <i>sUser</i>	The sub user account name	[in] xsd:string
Parameter 4 <i>sEmailCode</i>	Identity Code	[out] xsd:string
Parameter 5	Perform email filtering flag	[out] xsd:Boolean



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

<i>bEmailFilter</i>		
Parameter 6 <i>sEmailFilter1</i>	Email filter address 1	[out] xsd:string
Parameter 7 <i>sEmailFilter2</i>	Email filter address 2	[out] xsd:string
Parameter 8 <i>sEmailFilter3</i>	Email filter address 3	[out] xsd:string
Parameter 9 <i>bEmailSmsSplit</i>	Split long emails into multiple SMS flag	[out] xsd:Boolean
Parameter 10 <i>bEmailEndDoubleLine</i>	Terminate emails by double black lines flag	[out] xsd:Boolean
Parameter 11 <i>bEmailEndStop</i>	Terminate emails by special stop word ##END## flag	[out] xsd:Boolean
Parameter 12 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.22 Edit Email Messaging Configuration Settings (Partner Enabled Accounts)

This method allows partner account holders to edit the email messaging settings of the sub account holders.

partnerEditEmailMessaging		
Parameter 1 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 2 <i>sUser</i>	The sub user account name	[in] xsd:string
Parameter 3 <i>bEmailEnable</i>	Email messaging enabled flag	[in] xsd:Boolean
Parameter 4 <i>sEmailCode</i>	Identity Code	[in] xsd:string
Parameter 5 <i>bEmailFilter</i>	Perform email filtering flag	[in] xsd:Boolean
Parameter 6 <i>sEmailFilter1</i>	Email filter address 1	[in] xsd:string
Parameter 7 <i>sEmailFilter2</i>	Email filter address 2	[in] xsd:string
Parameter 8 <i>sEmailFilter3</i>	Email filter address 3	[in] xsd:string



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

Parameter 9 <i>bEmailSmsSplit</i>	Split long emails into multiple SMS flag	[in] xsd:Boolean
Parameter 10 <i>bEmailEndDoubleLine</i>	Terminate emails by double black lines flag	[in] xsd:Boolean
Parameter 11 <i>bEmailEndStop</i>	Terminate emails by special stop word ##END## flag	[in] xsd:Boolean
Parameter 12 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.23 Get Cost By Country (Partner Enabled Accounts)

This method allows partner account holders to determine the cost and wholesale cost by country. Please note that the actual cost algorithms and templates would be agreed programmed by messagingPeople and partner.

partnerGetCostByCountry		
Parameter 1 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer
Parameter 2 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 3 <i>sCountry</i>	The country of interest	[in] xsd:string
Parameter 4 <i>bMobileNumber</i>	Mobile service flag indicator	[in] xsd:boolean
Parameter 5 <i>sPackageType</i>	The package type to use	[in] xsd:string
Parameter 6 <i>sCurrency</i>	The currency type to use. There are 12 main currencies supported as shown in appendices.	[in] xsd:string
Parameter 7 <i>sServiceType</i>	The service type (SMS, MMS, WAP, FAX)	[in] xsd:string
Parameter 8 <i>dCostWholesaleFlat</i>	The wholesale flat cost	[out] xsd:double
Parameter 9 <i>dCostWholesalePerHalfMin</i>	The wholesale cost per half minute	[out] xsd:double
Parameter 10 <i>dCostResellerFlat</i>	The reseller flat cost	[out] xsd:double
Parameter 11 <i>dCostResellerPerHalfMin</i>	The reseller cost per half minute	[out] xsd:double



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

Parameter 12 <i>sNetworks</i>	The network overages supported	[out] xsd:string
----------------------------------	-----------------------------------	------------------

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

2.24 Get Cost By Phone Number (Partner Enabled Accounts)

This method allows partner account holders to determine the cost and wholesale cost by Phone Number. Please note that the actual cost algorithms and templates would be agreed programmed by messagingPeople and partner.

partnerGetCostByPhoneNumber		
Parameter 1 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer
Parameter 2 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 3 <i>sPhoneNumber</i>	The phone number	[in] xsd:string
Parameter 4 <i>sPackageType</i>	The package type to use	[in] xsd:string
Parameter 5 <i>sCurrency</i>	The currency type to use. There are 12 main currencies supported as shown in appendices.	[in] xsd:string
Parameter 6 <i>sServiceType</i>	The service type (SMS, MMS, WAP, FAX)	[in] xsd:string
Parameter 7 <i>dCostWholesaleFlat</i>	The wholesale flat cost	[out] xsd:double
Parameter 8 <i>dCostWholesalePerHalfMin</i>	The wholesale cost per half minute	[out] xsd:double
Parameter 9 <i>dCostResellerFlat</i>	The reseller flat cost	[out] xsd:double
Parameter 10 <i>dCostResellerPerHalfMin</i>	The reseller cost per half minute	[out] xsd:double
Parameter 11 <i>sCountry</i>	The country covered	[out] xsd:string
Parameter 12 <i>sNetworks</i>	The network overages supported	[out] xsd:string

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.25 Creating a Sub User Session (Partner Enabled Accounts)

There are times where the partner needs to perform operations and/or administrative tasks on behalf of the sub user (for example: obtain history details of sub users, account balance of sub users, send messages on behalf of sub user and so forth). This requirement is necessary for integrating into web portals development.



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

This method allows the session of the sub user belonging to the partner account to be returned which can be passed in to all the other APIs provided. Care should be taken so that you do *NOT* confuse this session with the session returned by the *sessionStart* method. The *sessionStart* method returns the session of the given user login whereas this method returns a session of the sub user belonging to the sub user account.

All sessions with the gateway are timed and are automatically disconnected after a period of idle activity or until the end session request is sent.

It is important that once you have finished you send the end session request to reclaim resources allocated.

partnerStartSessionForSubUser		
Parameter 1 <i>sUser</i>	Login	[in] xsd:string
Parameter 2 <i>sPassword</i>	Password	[in] xsd:string
Parameter 3 <i>sSubUser</i>	Sub User Account	[in] xsd:string
Parameter 4 <i>sSession</i>	Session	[out] xsd:string
Parameter 5 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.

2.26 Ending a Sub User Session (Partner Enabled Accounts)

Once interaction with the messagingPeople gateway is completed. Developers should end session to reclaim resources allocated.

partnerEndSessionForSubUser		
Parameter 1 <i>sSession</i>	Session Key	[in] xsd:string
Parameter 2 <i>nErrorCode</i>	Error Code (See Appendices 1)	[out] xsd:integer

Refer to Section 1.3 for the sample source code contained within the messagingPeople SDK installation package.



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

2.27 Partnership API

For all partnership APIs please refer to the Web Services documentation where this is extensively covered and supported.

Partnership APIs allow you to create and manage sub users. Resell and rebrand services applying your own margins and charges applied to the sub user accounts.

2.28 QR-Codes (Quick Response Codes) API

For all QR-Codes APIs please refer to the HTTP/HTTPS documentation where this is extensively covered and supported.



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

3. Appendices

3.1 Error Codes

ErrorCode	Description
0	Success
1	Unknown error
2	Bad arguments passed
3	User had some failed attempts to log in previously, forced delay
4	Bad user name or password
5	Can't create session
6	Exception error caught
7	No valid session
8	Database error
9	Security breach
10	Maximum number of contacts received for sending (break down into smaller batches)
11	Failed to call remote function over SOAP
12	Some file operation error occurred
13	Out of memory
14	Proxy settings error
15	Can not locate the messagingPeople Gateway Server
16	Setting schedule date failed (Possibly setting date too far in future)
17	Bad time format
18	Partnership operations is not enabled for this account
19	Partnership operation performed an illegal operation
20	Sub user account is not authorised for the given partner account
21	Sub user account is already in use – can not create this sub user account
22	Can not determine the cost at this point of time for the given settings passed in.

3.2 Currency Codes

Currency	Description
AUD	Australian Dollar
CAD	Canadian Dollar
CNY	Chinese Yuan



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

<http://helpdesk.messagingpeople.com>
<http://faq.messagingpeople.com>

messagingPeople

CHF	Swiss Franc
EUR	Euro
GBP	British Pound
HKD	Hong Kong Dollar
JPY	Japanese Yen
MYR	Malaysian Ringgit
NZD	New Zealand Dollar
SGD	Singaporean Dollar
USD	USA Dollar

3.3 Package Types

Package	Description
STANDARD	Standard package
FREQUENT	Frequent package
BULK1	Bulk 1 package
BULK2	Bulk 2 package
BULK3	Bulk 3 package
BULK4	Bulk 4 package
BULK5	Bulk 5 package