



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

messagingPeople Developer APIs for COM

Version 2.08 (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: <http://helpdesk.messagingpeople.com> or visit the Frequently Asked Questions list (FAQ) at <http://faq.messagingpeople.com>.



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

Table of Contents

1. HISTORY	3
2. GENERAL.....	4
2.1 SUPPORTED ENVIRONMENTS.....	4
2.2 SAMPLE SOURCE CODE.....	4
2.3 GETTING STARTED.....	4
3. DEVELOPERS APIS	6
3.1 INSTANTIATING THE MESSAGINGPEOPLE COM OBJECT	6
3.2 PROXY CONNECTIONS.....	7
3.3 CREATING A SESSION.....	8
3.4 ENDING A SESSION.....	10
3.5 GET VERSION.....	11
3.6 GET ACCOUNT BALANCE.....	12
3.7 GET MESSAGINGPEOPLE NEWS	13
3.8 START SCHEDULING MESSAGES	14
3.9 STOP SCHEDULING MESSAGES.....	15
3.10 SEND SMS	17
<i>Plain text SMS:</i>	18
<i>Unicode SMS:</i>	18
3.11 SEND MMS.....	21
3.12 SEND WAP	22
3.13 SEND FAX.....	23
3.14 GET MESSAGE STATUS	24
3.15 SET SENDER ID.....	25
3.16 PARTNERSHIP API.....	26
3.17 QR-CODES (QUICK RESPONSE CODES) API	26
4. APPENDICES	27
4.1 ERROR CODES.....	27



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

1. History

Date	Description
01 Aug 2006	Initial release.
27 Sep 2006	Updated the document logo.
09 Aug 2008	Added partnership and QR-Codes API information.
22 Dec 2008	Added long SMS and Unicode SMS details.
15 May 2010	Updated documentation templates and new method setSenderId
17 Dec 2011	Removed deprecated functionality



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

2. General

2.1 Supported Environments

COM technology was built to operate exclusively under the Windows Operating System. As such the use of the COM developer APIs can only be used under Microsoft Windows platforms.

Other Operating System platforms can integrate into the messagingPeople gateway via other interfaces (HTTP/HTTPS, SMTP and WSDL). Please refer to our website for details on different interfaces supported.

You can use the messagingPeople COM within IIS (ASP/ASP.NET), Apache (versions that support COM under Windows), VB, .NET, VC++ and other development environments that directly support COM.

2.2 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>

2.3 Getting Started

Download and install the messagingPeople COM downloadable from our website at <http://www.messagingpeople.com>. The installation simply deploys the COM gateway and documentation. It also registers your COM using the command:

```
regsvr32 messagingPeopleGateway.dll
```

If your writing applications that use the messagingPeople COM API then you will need to ensure that these are also registered on the target computers as part of the deployment process.

Some computers may be required to set the COM permissions for allowing the messagingPeople gateway COM to be instantiated and launched. Failure to do this results in a permission denied error when instantiating the messagingPeople gateway COM.



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

Permissions can be set by directly setting the permissions on the messagingPeople Gateway using the Dcomcnfg or OLE/OLE Object Viewer applications. Please contact support at support@messagingpeople.com for any questions.



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

3. Developers APIs

3.1 Instantiating the messagingPeople COM Object

Before you can instantiate the messagingPeople COM you will need to ensure that the COM object has already been registered via the regsvr32 command. Access to the messagingPeople APIs only become exposed after instantiating the COM.

Various different development environments have different ways in instantiating the COM object. Some examples have been provided below:

Example:

Refer to Section 2.2 for the sample source code contained within the messagingPeople SDK installation package.

Apache:

```
<?php
// PHP has a COM and non COM solutions. See SDK for details

// Create COM instance
$comMP = new COM("MessagingPeopleGateway.MessagingPeopleGatewayInterface.1") or
exit("Failed");

// Use the COM

// Free COM resources
$comMP = null;
?>
```

ASP:

```
<%
\ Create COM instance
Dim comMP
comMP = CreateObject("MessagingPeopleGateway.MessagingPeopleGatewayInterface.1")

\ Use the COM

\ Free the COM
Set comMP = Nothing
%>
```



PO BOX 809
 Glen Waverley
 VIC 3150
 AUSTRALIA

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

messagingPeople

VB/VB.NET:

```
` Add the COM into the project and directly start using
Dim comMP As MessagingPeopleGateway
```

```
` Use the COM
```

```
` Free the COM
Set comMP = Nothing
```

C++:

```
// Make sure you point to the exact location to where the messagingPeople Gateway
// is located. The COM header file wrappers will be generated automatically.
// Simply include these into your project.
#import "C:\Program Files\messagingPeople\messagingPeopleGateway.dll"
```

```
IMessagingPeopleGatewayPtr comMP;
HRESULT hResult = comMP.CreateInstance(__uuidof(MessagingPeopleGateway));
if (hResult != S_OK)
{
    // Error
}
```

```
// Use the COM
```

```
// Free the COM
comMP = NULL;
```

3.2 Proxy Connections

You do not have to worry about proxy connections if you connect direct to the Internet without going via the proxy server. By default the COM assumes direct Internet connections.

However, if you are behind the proxy server then your required to provide the user name, password, proxy host and proxy port details. Without these details you will not be able to pass past the proxy server to connect to the messagingPeople gateway.

setProxy		
Parameter 1	Proxy User Name	[in] String
Parameter 2	Proxy Password	[in] String
Parameter 3	Proxy Host	[in] String
Parameter 4	Proxy Port Number	[in] Integer



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

Example:

Refer to Section 2.2 for the sample source code contained within the messagingPeople SDK installation package.

Apache:

```
<?php
// PHP has a COM and non COM solutions. See SDK for details

// Set the details of your proxy user name, password, host and port number
$comMP->setProxy("<<username>>", "<<password>>", "<<proxyhost>>", 8080);
?>
```

ASP:

```
<%
` Set the details of your proxy user name, password, host and port number
comMP.setProxy("<<username>>", "<<password>>", "<<proxyhost>>", 8080)
%>
```

VB/VB.NET:

```
` Set the details of your proxy user name, password, host and port number
comMP.setProxy("<<username>>", "<<password>>", "<<proxyhost>>", 8080)
```

C++:

```
// Set the details of your proxy user name, password, host and port number
HRESULT hResult
    = comMP->setProxy("<<username>>", "<<password>>", "<<proxyhost>>", 8080)
if (hResult != S_OK)
{
    // Error
}
```

3.3 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.



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

sessionStart		
Parameter 1	Login	[in] String
Parameter 2	Password	[in] String
Parameter 3	Session	[out] String / Variant
Parameter 4	Error Code (See Appendices 1)	[out] Integer / Variant

Example:

Refer to Section 2.2 for the sample source code contained within the messagingPeople SDK installation package.

Apache:

```
<?php
// PHP has a COM and non COM solutions. See SDK for details

$session = new VARIANT();
$errorcode = new VARIANT();

// Create a session
$comMP->sessionStart("<<login>>", "<<password>>", $session, $errorcode);
?>
```

ASP:

```
<%
\ Create a session
Dim session As String
Dim errorcode As Integer
comMP.sessionStart("<<login>>", "<<password>>", session, errorcode)
%>
```

VB/VB.NET:

```
\ Create a session
Dim session As String
Dim errorcode As Integer
comMP.sessionStart("<<login>>", "<<password>>", session, errorcode)
```

C++:

```
// Create a session
VARIANT session;
VARIANT errorcode;
HRESULT hResult
```



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

```
    = comMP->sessionStart("<<login>>", "<<password>>", &session, &errorcode)  
if (hResult != S_OK)  
{  
    // Error  
}
```

3.4 Ending a Session

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

sessionEnd		
Parameter 1	Session Key	[in] String
Parameter 2	Error Code (See Appendices 1)	[out] Integer / Variant

Example:

Refer to Section 2.2 for the sample source code contained within the messagingPeople SDK installation package.

Apache:

```
<?php  
  
// PHP has a COM and non COM solutions. See SDK for details  
  
// End session  
$comMP->sessionEnd($session, $errorcode);  
?>
```

ASP:

```
<%  
` End session  
comMP.sessionEnd(session, errorcode)  
%>
```

VB/VB.NET:

```
` End session  
comMP.sessionStart(session, errorcode)
```

C++:

```
// End session
```



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

```
HRESULT hResult  
    = comMP->sessionEnd(session, &errorcode)  
if (hResult != S_OK)  
{  
    // Error  
}
```

3.5 Get Version

Retrieves the version of your messagingPeople gateway COM object.

getVersion		
Parameter 1	Version Number	[out] Integer / Variant

Example:

Refer to Section 2.2 for the sample source code contained within the messagingPeople SDK installation package.

Apache:

```
<?php  
  
// PHP has a COM and non COM solutions. See SDK for details  
  
$version = new VARIANT();  
  
// Get version  
$comMP->getVersion($version);  
?>
```

ASP:

```
<%  
' Get version  
Dim version As Integer  
comMP.getVersion(version)  
%>
```

VB/VB.NET:

```
' Set the details of your proxy user name, password, host and port number  
Dim version As Integer  
comMP.getVersion(version)
```

C++:



PO BOX 809
 Glen Waverley
 VIC 3150
 AUSTRALIA

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

messagingPeople

```
// Get version
VARIANT version;
HRESULT hResult
    = comMP->getVersion(&version)
if (hResult != S_OK)
{
    // Error
}
```

3.6 Get Account Balance

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

getAccountBalance		
Parameter 1	Session Key	[in] String
Parameter 2	Balance	[out] Double / Variant
Parameter 3	Error Code (See Appendices 1)	[out] Integer / Variant

Example:

Refer to Section 2.2 for the sample source code contained within the messagingPeople SDK installation package.

Apache:

```
<?php
// PHP has a COM and non COM solutions. See SDK for details

$balance = new VARIANT();
$errorcode = new VARIANT();

// Gets account balance
$comMP->getAccountBalance($session, $balance, $errorcode);
?>
```

ASP:

```
<%
' Gets the account balance
Dim balance As String
Dim errorcode As Integer
comMP.getAccountBalance(session, balance, errorcode)
%>
```

VB/VB.NET:



PO BOX 809
 Glen Waverley
 VIC 3150
 AUSTRALIA

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

messagingPeople

```
` Gets the account balance
Dim balance As String
Dim errorcode As Integer
comMP.getAccountBalance(session, balance, errorcode)
```

C++:

```
// Gets the account balance
VARIANT balance;
VARIANT errorcode;
HRESULT hResult
    = comMP->getAccountBalance(&session, &balance, &errorcode)
if (hResult != S_OK)
{
    // Error
}
```

3.7 Get messagingPeople News

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

getNews		
Parameter 1	News	[out] String / Variant
Parameter 2	Error Code (See Appendices 1)	[out] Integer / Variant

Example:

Refer to Section 2.2 for the sample source code contained within the messagingPeople SDK installation package.

Apache:

```
<?php
// PHP has a COM and non COM solutions. See SDK for details

$news = new VARIANT();
$errorcode = new VARIANT();

// Gets news
$comMP->getNews($news, $errorcode);
?>
```

ASP:



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

```
<%
  \ Gets the news
Dim news As String
Dim errorcode As Integer
comMP.getNews(news, errorcode)
%>
```

VB/VB.NET:

```
\ Gets the news
Dim news As String
Dim errorcode As Integer
comMP.getNews(news, errorcode)
```

C++:

```
// Gets the news
VARIANT news;
VARIANT errorcode;
HRESULT hResult
    = comMP->getNews(&session, &news, &errorcode)
if (hResult != S_OK)
{
    // Error
}
```

3.8 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.

scheduleEnable		
Parameter 1	Session Key	[in] String
Parameter 2	The date/time in GMT to send the message. Must be in the format as follows: "YYYYmdd+HHMM" where YYYY is the year, mm is the month, dd is the day, HH is the hour and MM is the minutes. HH range is [0, 24).	[in] String / Variant
Parameter 3	Error Code (See Appendices 1)	[out] Integer / Variant

Example:



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

Refer to Section 2.2 for the sample source code contained within the messagingPeople SDK installation package.

Apache:

```
<?php
// PHP has a COM and non COM solutions. See SDK for details

$schedule = "20060913+1430";
$errorcode = new VARIANT();

// Starts scheduling
$comMP->scheduleEnable($session, $schedule, $errorcode);
?>
```

ASP:

```
<%
` Start scheduling
Dim schedule As String
Dim errorcode As Integer
comMP.scheduleEnable(session, schedule, errorcode)
%>
```

VB/VB.NET:

```
` Starts scheduling
Dim schedule As String
Dim errorcode As Integer
comMP.scheduleEnable(session, schedule, errorcode)
```

C++:

```
// Starts scheduling
VARIANT schedule;
VARIANT errorcode;
HRESULT hResult
    = comMP->scheduleEnable(&session, &schedule, &errorcode)
if (hResult != S_OK)
{
    // Error
}
```

3.9 Stop Scheduling Messages

Stops scheduling messages enabled via the Start Scheduling Messages.



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

scheduleDisable		
Parameter 1	Session Key	[in] String
Parameter 2	Error Code (See Appendices 1)	[out] Integer / Variant

Example:

Refer to Section 2.2 for the sample source code contained within the messagingPeople SDK installation package.

Apache:

```
<?php
// PHP has a COM and non COM solutions. See SDK for details

$errorcode = new VARIANT();

// Disable scheduling
$comMP->scheduleDisable($session, $errorcode);
?>
```

ASP:

```
<%
` Disable scheduling
Dim errorcode As Integer
comMP.scheduleDisable(session, errorcode)
%>
```

VB/VB.NET:

```
` Disable scheduling
Dim errorcode As Integer
comMP.scheduleDisable(session, errorcode)
```

C++:

```
// Disable scheduling
VARIANT errorcode;
HRESULT hResult
    = comMP->scheduleDisable(&session, &errorcode)
if (hResult != S_OK)
{
    // Error
}
```



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

3.10 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	Session Key	[in] String
Parameter 2	Message for plain text or Unicode message. See details above.	[in] String
Parameter 3	Flash	[in] Boolean
Parameter 4	List of Phone Numbers (must be in international format)	[in] String
Parameter 5	List of Receipts	[out] String / Variant
Parameter 6	Error Code (See Appendices 1)	[out] Integer / Variant

Example:

Refer to Section 2.2 for the sample source code contained within the messagingPeople SDK installation package.

Apache:

```
<?php
// PHP has a COM and non COM solutions. See SDK for details

$receipts = new VARIANT();
$errorcode = new VARIANT();
$message1 = "My SMS message";
$message2 = "%UNICODE%4F60597D";
$flash = false;

// Send SMS
$comMP->sendSms($session, $message1, $flash, $receipts, $errorcode);
$comMP->sendSms($session, $message2, $flash, $receipts, $errorcode);
?>
```



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

ASP:

```
<%  
' Send SMS  
Dim receipts As String  
Dim errorcode As Integer  
Dim message As String  
message1 = "My SMS message"  
message2 = "%UNICODE%4F60597D"  
Dim flash As Boolean  
Flash = False  
comMP.sendSms(session, message1, flash, receipts, errorcode)  
comMP.sendSms(session, message2, flash, receipts, errorcode)  
>
```

VB/VB.NET:

```
' Send SMS  
Dim receipts As String  
Dim errorcode As Integer  
Dim message1 As String  
message1 = "My SMS message"  
message2 = "%UNICODE%4F60597D"  
Dim flash As Boolean  
Flash = False  
comMP.sendSms(session, message1, flash, receipts, errorcode)  
comMP.sendSms(session, message2, flash, receipts, errorcode)
```

C++:

```
// Send SMS  
VARIANT receipts;  
VARIANT errorcode;  
_bstr_t oBstrMessage1 = "My SMS message";  
_bstr_t oBstrMessage2 = "%UNICODE%4F60597D";  
bool flash = false;  
HRESULT hResult  
= comMP->sessionStart(session, oBstrMessage1, flash, &receipts, &errorcode)  
if (hResult != S_OK)  
{  
    // Error  
}  
hResult  
= comMP->sessionStart(session, oBstrMessage2, flash, &receipts, &errorcode)  
if (hResult != S_OK)  
{  
    // Error  
}
```



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

3.11 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. The content is loaded directly from the file passed into the COM interface. Thus you will need to ensure that the file is not locked and adequate access permissions are set.

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	Session Key	[in] String
Parameter 2	Description (Keep this empty for no description)	[in] String
Parameter 3	File Name	[in] String
Parameter 4	List of Phone Numbers (must be in international	[in] String



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

	format)	
Parameter 5	List of Receipts	[out] String / Variant
Parameter 6	Error Code (See Appendices 1)	[out] Integer / Variant

Example of code is similar as shown for the sendSMS example.

Refer to Section 2.2 for the sample source code contained within the messagingPeople SDK installation package.

3.12 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.

The content for the WAP is loaded directly from the file passed into the COM interface. Thus you will need to ensure that the file is not locked and adequate access permissions are set.

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):



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

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

SendWap		
Parameter 1	Session Key	[in] String
Parameter 2	Message	[in] String
Parameter 3	File Name	[in] String
Parameter 4	List of Phone Numbers (must be in international format)	[in] String
Parameter 5	List of Receipts	[out] String / Variant
Parameter 6	Error Code (See Appendices 1)	[out] Integer / Variant

Example of code is similar as shown for the sendSMS example.

Refer to Section 2.2 for the sample source code contained within the messagingPeople SDK installation package.

3.13 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.

The content for the FAX is loaded directly from the file passed into the COM interface. Thus you will need to ensure that the file is not locked and adequate access permissions are set.

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:



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

+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	Session Key	[in] String
Parameter 2	File Name	[in] String
Parameter 3	List of Phone Numbers (must be in international format)	[in] String
Parameter 4	List of Receipts	[out] String / Variant
Parameter 5	Error Code (See Appendices 1)	[out] Integer / Variant

Example of code is similar as shown for the sendSMS example.

Refer to Section 2.2 for the sample source code contained within the messagingPeople SDK installation package.

3.14 Get Message Status

This method gets the Status of the message sent.

getStatus		
Parameter 1	Session Key	[in] String
Parameter 2	Receipt	[in] String
Parameter 3	Status of the message	[out] String / Variant
Parameter 4	Error Code (See Appendices 1)	[out] Integer / Variant

Example:

Refer to Section 2.2 for the sample source code contained within the messagingPeople SDK installation package.

Apache:

```
<?php
```



PO BOX 809
 Glen Waverley
 VIC 3150
 AUSTRALIA

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

messagingPeople

```
// PHP has a COM and non COM solutions. See SDK for details

$status = new VARIANT();
$errorcode = new VARIANT();

// Get status
$comMP->getStatus($session, $receipt, $status, $errorcode);
?>
```

ASP:

```
<%
  ` Get status
Dim Status As String
Dim errorcode As Integer
comMP.getStatus(session, receipt, status, errorcode)
%>
```

VB/VB.NET:

```
` Get status
Dim Status As String
Dim errorcode As Integer
comMP.getStatus(session, receipt, status, errorcode)
```

C++:

```
// Send SMS
VARIANT status;
VARIANT errorcode;
HRESULT hResult
    = comMP->getStatus(session, receipt, &status, &errorcode)
if (hResult != S_OK)
{
    // Error
}
```

3.15 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.

setSenderId		
Parameter 1	Session Key	[in] String
Parameter 2	Sender ID	[in] String
Parameter 3	Error Code (See Appendices	[out] Integer / Variant



PO BOX 809
Glen Waverley
VIC 3150
AUSTRALIA

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

messagingPeople

	1)	
--	----	--

Example:

Refer to Section 2.2 for the sample source code contained within the messagingPeople SDK installation package.

3.16 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.

3.17 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

4. Appendices

4.1 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	Can not locate the messagingPeople Gateway Server
15	Setting schedule date failed (Possibly setting date too far in future)
16	Bad time format
17	Partnership account not authorised
18	Partnership functionality performed illegal operation
19	Partnership sub-account not authorised
20	Partnership sub-account already in use
21	Partnership can not calculate the cost
22	Deprecated functionality