



TECHNICAL DOCUMENTATION
FOR:

SOAP Web Management Services

TextPower, Inc.
www.TextPower.com
[Twitter.com/TextPower](https://twitter.com/TextPower)
Support@TextPower.com
888-818-1808

Proprietary & Confidential Information
Copyright © 2017



SOAP Web Management Services

This manual describes the Version 2 SOAP interface to TextPower’s Web Management Services. There are multiple levels of SOAP interfaces. Only the Web Management Services Interface is described in this manual. Access to the SOAP interfaces does not come automatically with TextPower service. Access to each SOAP level must be specifically requested. Contact your account representative.

In this manual, the term MT means Message Terminating. That means a message that you send to a phone. The term MO means Message Originating. That means a message that a phone sends to you. The terms terminating and originating always are from the perspective of the phone. TextPower now also supports Email sending. For Email sending, different standards apply.

Table of Contents

Table of Contents	1
Overview	2
WSDL and Header	2
Function Calls	3
OptIn Interface for web sites.....	3
Process Flow for Web Based OptIn:.....	4
Cell Numbers	4
Email Addresses.....	4
Division of Responsibility	4
Application Design	4
Method 1	4
Method 2	5
OptIn Logic Flow.....	5
Subscription Management	6
Subscription Features.....	6
Subscription Mechanics	7
SendPassCode	8
ManageOptIn	9
SetAutoWelcome	12
SendAutoWelcome	14
SetOptinTimeout.....	15
ManageSubscription	17

SOAP Web Management Services

Overview

This manual will not attempt to describe the general SOAP interface and structure. It is expected that the reader already knows this. This manual provides only supplementary information that is not found in the WSDL descriptions. This interface along with the other SOAP interfaces allows the TextPower (TPI) customer to build extremely sophisticated messaging applications. Building a SOAP interface should not be undertaken unless the developer has the proper development tools to auto-generate the necessary interface code and the expertise to develop the SOAP application.

All TextPower Web Management SOAP functions return an XML node. A SOAP header is required on all calls and the header information is identical on all calls. Structural errors involving invalid credentials, incorrect parameter values, etc. return an XML fragment with Errors as the base node. SOAP exceptions are supported. SOAP exceptions always represent internal program faults. Hopefully, you will not receive any but if you do, please forward the information to TextPower support. The return of an Errors node represents a detectable error in the information supplied. Please check the information that you are supplying to correct this type of problem.

Standard TextPower accounts do not come with the SOAP interface enabled. Users must have specific permission to use each of the provided SOAP interfaces. Contact your account representative to see if your proposed application can be approved to use SOAP.

WSDL and Header

The WSDL document for the Web Management SOAP interface is located at <http://www.textpower.com/TPIServices/WebManagementServicesV2.asmx?wsdl>

or

<https://secure.textpower.com/TPIServices/WebManagementServicesV2.asmx?wsdl>

The address for the SOAP calls on the advanced interface is, of course:

<http://www.textpower.com/TPIServices/WebManagementServicesV2.asmx>

or

<https://secure.textpower.com/TPIServices/WebManagementServicesV2.asmx>

Note: This version of the Web Management SOAP interface document provides better access for PHP users. For other users, the features are unchanged.

The SOAP header on all calls has four parameters as described in the WSDL. All header parameters are of type String.

SOAP Web Management Services

Parameter	Meaning
UID	Required. The value of your userid as assigned by TextPower.
PWD	Required. The value of your password as assigned by TextPower.
Campaign	The name of the campaign you are using for this call. If you have only one campaign, this parameter may be a zero length string.
Keyword	The keyword in your campaign that you are using for this call. If you have only one keyword in your campaign, this parameter may be zero length string.

The majority of users has only one campaign each and only one keyword and will be able to leave the Campaign and Keyword parameters alone.

Function Calls

Type specifications for each call and the parameter data types must be taken from the WSDL. The basic purpose of each function is also described in the WSDL. This section provides additional information on the meaning of each call, its parameters and return values along with an example of the returned XML node.

Note that in the WSDL, the Integer type is referred to as int, the Date type is referred to as dateTime and the XmlNode type is a complex type. They are the same thing and may be used interchangeably. Sample output XML has been formatted for display on this page using line feeds and indents. These do not appear in actual XML sent from the switch. The Level parameter for output nodes denotes the tag nesting level. The root or base node is 1.

Date and Time formats: Note that all dates and times in the Web Management specification of both inputs and outputs are expressed in Universal Time, except where noted.

OptIn Interface for web sites

When short codes and keywords require OptIn, some campaigns may be certified for using the web as an OptIn method. Using the web requires that the user enter their cell phone number and have displayed to them the same OptIn A message that they would see on their phone. You may customize this message for the web by eliminating things that apply only to cell phones. An example might be changing a statement such as “*Reply Yes to Accept*”. After this, they may be considered opted in and messages may be sent to them according to the terms of the OptIn A message.

TextPower keeps the OptIn information in its own database. External applications where TextPower acts only as a gateway, need a way to create web based OptIns. That is what



SOAP Web Management Services

this interface does. OptIn now supports both Cell Numbers for SMS sending and Emails for Email Sending.

Process Flow for Web Based OptIn:

Cell Numbers

1. User is directed to the applicable web page.
2. User enters cell phone number.
3. User has displayed to them the terms of the sign up.
4. User is sent a special pass code to the phone that they wish to OptIn on.
5. User enters the pass code received on their phone into the web page. If the entered pass code and the transmitted pass code are the same, the user is Opted In. The pass code makes sure that the web user is in physical possession of the cell phone that is to be Opted In and prevents fraud or massive unauthorized OptIns of users.
6. User is given acknowledgement of the result of the transaction via a carrier compliant text message (Welcome Message).

Email Addresses

1. User is directed to the applicable web page.
2. User enters email address.
3. User has displayed to them the terms of the sign up.
4. User is sent a special email to the email address that they wish to OptIn on.
5. User clicks on the confirm link in the email to complete the OptIn.

Division of Responsibility

The application will be responsible for taking care of steps 1, 2 and 6 above. Steps 2, 4 and 5 require access to the TextPower OptIn database and must be taken care of by TextPower by calling functions detailed in this document.

The application interface is housed on a web page of the application's choosing that meets the interface standards set forth later in this document.

Application Design

Since direct access to the OptIn database is restricted to TextPower code, the user will need these primitives to access the database. There are two methods to accomplish this.

Method 1

House the OptIn code on TextPower's servers as in the example. Custom OptIn pages can be created for customers although there will be a charge for this. Customers often want their own domain name to show in the browser address bar however. You

SOAP Web Management Services

potentially could host your registration page on your own server with your own domain and put the Actual OptIn code in a frame that references a TextPower domain. Due to cross-domain security issues, there are very few browsers anymore that will permit the necessary references across domains. If you wish to take this route, you should be advised that the browser address bar would have to show a TextPower domain as in the example.

Method 2

As an application, you may host an OptIn page on your own domain and provide all the necessary logic there. To gain the access you need to create the actual OptIn, you will need to use the primitives supplied by TextPower. You must use the SOAP calls described in this document.

OptIn Logic Flow

The CTIA approved method for doing web OptIns for cell numbers is summarized below. This discussion does not apply to email addresses. You must follow this method to have your application certified by the carriers.

1. Collect the Cell Phone Number from the user on your web page. The carrier associated with the cell phone number will always be looked up automatically by TextPower. There is no need to collect it on the web site.
2. Display the terms such as charges and messages that they may receive to the customer. They must click on a button that indicates their agreement to those terms.
3. Send a Pass Code to the phone attempting to be registered. This is necessary to verify that the person on the web page is actually in possession of the phone that is being registered.
4. Have the person enter on your web page the pass code that they received.
5. If the pass code matches, you may Opt them in.

Of course, you should handle all the error conditions that can result from this process.

You will need to use these three primitives from TextPower to accomplish these tasks. The interfaces for these primitives are documented herein.

1. The ManageOptin call allows you to set, reset and query the state of a user's optin to your keywords and short code.
2. The SendPassCode call allows you to send a pass code to a user. You cannot use TextPower's regular sending primitives for this function since they will check the OptIn state before allowing a send. By definition, you don't have an OptIn at this point.
3. Send the required, carrier compliant welcome message to the new subscriber. This may be done after OptIn using the user's own logic or you may use the AutoWelcome feature of the TextPower system to send this message. This is controlled by the SetAutoWelcome and SendAutoWelcome primitives described in this document.

Subscription Management

TextPower provides subscription services on its switch. By definition, almost all subscription services are Premium Rated requiring a monthly fee from each consumer who subscribes to the service. There are some specific uses of “free” subscription services such as a closed community alerting service. Subscription services can be initiated from either an MO or from a web interface. The most common scenario for initiating subscription services however is from a web site and this document will solely address that method. For information on MO initiated subscription services, contact your account representative.

Subscription Features

1. Subscription packages normally carry a monthly fee although the fee may be zero. TextPower bills the customer if the fee is non-zero through their carrier for the initial month and will bill the customer at 7PM Central time each month thereafter on same date of the month in which they signed up. If a subsequent month has fewer days than the month in which the signup took place, they will be billed on the first of the following month. Revenue sharing takes place with the customer on a basis agreed upon with TextPower.
2. While a member of a subscription package, no messages or content sent to the consumer are charged a fee. The essence is that if the consumer pays a flat monthly fee, their usage for the month is at no additional charge. The TextPower customer however may set a limit on the number of content downloads that a consumer may execute for that month.
3. Customers may send bulk text messages or content to subscribing consumers subject to the limits communicated to the subscribers when they joined.
4. Customers may specify content that is available to each subscription package.
5. Customer may mark content as exclusive to a subscription package.
6. Consumers may OptOut (unsubscribe) to the service at any time. No refund is provided for a partial month left on the subscription.
7. TextPower Subscription Services take care of getting the customer into the subscription package along with the underlying billing support, rebilling the customer on each monthly anniversary of their signup and of getting them out of the package along with the necessary billing cancellation work with the carriers. While they are in the package however, the application gets all MO messages, if requested, from the user and takes care of sending them whatever content or messages they desire.
8. If any subscription member has any bill rejected for any reason, the subscriber will be removed from the applicable subscription package. Since billing rejections may not be delivered for up to 48 hours, this rejection may not take place immediately.



SOAP Web Management Services

Subscription Mechanics

Customers may create any number of subscription packages. These packages must be defined through the customer's login to the TextPower web site. Each subscription package has a unique identifier called a Subscription ID number. Each subscription package must also be linked to a unique short code/keyword pair. If the consumer signs up for their subscription and requests their content solely from a web site, they will never know what the keyword is. They will see the short code only as the source of their text messages or content. Since the keyword for web-based subscriptions is never seen or entered by the consumer, TextPower suggests that web only keywords start with *subscr* to distinguish them from regular keywords.

Unless the subscription fee is zero, the keyword will be subject to double OptIn. You may not deliver premium content both on a per item charge basis and from a subscription on the same short code/keyword pair. If you are entering a number into a subscription from a web site, you must set the optin before attempting to set the subscription.

SOAP Web Management Services

SendPassCode

SendPassCode (CellNumber As String, Carrier As String, ServiceName As String) As XmlNode

Send a pass code to the user to verify that they are in possession of a particular phone.

The SMS message sent to the user will be;

Your Pass Code for {ServiceName} is nnnn.

{ServiceName} will be replaced with what you sent to us in the Service Name parameter.

Nnnn will be the Pass Code generated by TextPower. It will be either 4 or 5 digits. You do not need to send Pass Codes to Email subscribers.

Input Parameters		
Name	Data Type	Meaning
CellNumber	String	The number of the cell phone to which the message is to be sent. The cell number must be exactly 10 digits for North American traffic.
Carrier	String	The carrier of the Cell Number. You should always supply the carrier if you have it. If you do not supply it, there may be a charge for the lookup. This parameter should be either a valid carrier code or the empty string (""). If it is the empty string, the true carrier will be returned to you in the output.
ServiceName	String	An acceptable name for your service under CTIA/MMA/Carrier rules.
Output Base Node:		PassCodeInfo
Sub Nodes		
Tag	Level-Type	Meaning
PassCode	2-Integer	The pass code generated and sent to the client
Carrier	2-String	The carrier code. If a valid carrier code was supplied on input, it is simply echoed in this output parameter. If the empty string was supplied, the true carrier is looked up and returned.
Output Example		
<pre><PassCodeInfo xmlns=""> <PassCode>52755</PassCode> <Carrier>00000</Carrier> </PassCodeInfo></pre>		

SOAP Web Management Services

ManageOptIn

Function ManageOptIn(Action As String, CellNumber As String, Carrier As String) As XmlNode

Manage the OptIn function for a number when subscribing a customer from a web site. Note that if the campaign has the Force OptIn attribute set and the cell number has previously Opted Out for whatever reason, the Manage Optin call will fail(see example 5). The number can only be re-OptedIn via an MO optin.

Input Parameters		
Name	Data Type	Meaning
Action	String	The OptIn action to be performed. Allowable case insensitive values are: Set- create an OptIn Reset – remove an OptIn Query – Get the current OptIn state
CellNumber	String	The number of the cell phone or email address that is to be acted upon for the OptIn action. A cell number must be exactly 10 digits for North American traffic. An email address must be well formed.
Carrier	String	The carrier of the Cell Number. If you do not supply the carrier, it will be looked up. For email addresses, use carrier="00000". Note: <i>If you are using a fictitious movie number, you must supply a carrier ID. It does not make any difference what carrier you supply. The carrier is just as fictitious as the number.</i>
Output Base Node:		OptIn
Sub Nodes		
Tag	Level-Type	Meaning
TimeStamp	2-Date	The date of the information retrieval or action
Action	2-String	The action that was performed
CellNumber/Email	2-String	The cell number or email address on which the action was performed. If the action was performed on a Cell Number, the tag is named CellNumber. If the action was performed on an Email Address, the tag is named Email.
Result	2-String	The result of the requested action. Possible results: For command set: <ul style="list-style-type: none"> • Campaign is not marked for OptIn

SOAP Web Management Services

		<ul style="list-style-type: none"> • OptIn Created • Cannot Force Re-Optin <p>For command reset:</p> <ul style="list-style-type: none"> • Optin not active: No Action • OptIn Deleted <p>For command query</p> <ul style="list-style-type: none"> • No OptIn • Waiting for Yes • Opted In <p>Error-(code)</p>
Output Example 1		
<pre><OptIn xmlns=""> <TimeStamp>2/25/2008 11:43:00 AM</TimeStamp> <Action>query</Action> <CellNumber>6345550144</CellNumber> <Result>No OptIn</Result> </OptIn></pre>		
Output Example 2a: Create a Cell Number Optin		
<pre><OptIn xmlns=""> <TimeStamp>2/25/2008 11:52:05 AM</TimeStamp> <Action>set</Action> <CellNumber>6345550144</CellNumber> <Result>OptIn Created</Result> </OptIn></pre>		
Output Example 2b: Create an Email Address Optin		
<pre><OptIn xmlns=""> <TimeStamp>1/20/2010 2:49:53 AM</TimeStamp> <Action>set</Action> <Email>myEmail@MyEmail.com</Email> <Result>OptIn Created</Result></OptIn></pre>		
Output Example 3		
<pre><OptIn xmlns=""> <TimeStamp>2/25/2008 11:53:04 AM</TimeStamp> <Action>reset</Action> <CellNumber>6345550144</CellNumber> <Result>OptIn Deleted</Result> </OptIn></pre>		
Output Example 4 -Attempt to reset an OptIn that does not exist		
<pre><OptIn xmlns=""> <TimeStamp>2/25/2008 11:50:17 AM</TimeStamp> <Action>reset</Action> <CellNumber>6345550144</CellNumber></pre>		

SOAP Web Management Services

```
<Result>No Optin Active: No Action</Result>  
</OptIn>
```

Output Example 5 -Attempt to set an OptIn that has been previously Opted Out and the account has the Force OptIn attribute set

```
<OptIn xmlns="">  
<TimeStamp>2/25/2008 11:52:05 AM</TimeStamp>  
<Action>set</Action>  
<CellNumber>6345550144</CellNumber>  
<Result>Cannot Re-force Optin</Result>  
</OptIn>
```

SetAutoWelcome

Function SetAutoWelcome(**ByVal** Message **As** String, **ByVal** OptInLimit **As** Integer, **ByVal** Clear **As** Boolean) **As** XmlNode

This function sets the parameters on the TextPower AutoWelcome feature for the short code and keyword attached to the credentials you provided. In an MO based registration, you may set up your account to issue the carrier mandated welcome message upon the initial OptIn of a particular cell phone to the short code/keyword combination. This primitive allows the user to control their auto welcome. Once set via this primitive, you may use the information to send a welcome MT using the SendAutoWelcome primitive described elsewhere in this document or you may use this primitive to set the information for normal MO based OptIns.

Input Parameters		
Name	Data Type	Meaning
Message	String	The carrier compliant welcome message that will be sent to subscribers when they first OptIn. This message needs to be set only once for each Short Code/Keyword combination.
MOLimit	Integer	The limit for MOs that will be accepted for each cell number. This parameter is usually used to limit entries allowed for contests. A value of zero means no limit.
Clear	Boolean	If True, the Auto Welcome feature will be cleared. If False, it will be set.
Output Base Node:		OptIn
Sub Nodes		
Tag	Level-Type	Meaning
TimeStamp	2-Date	The date of the action
Action	2-String	The action that was performed
Result	2-String	The result of the requested action. Possible results: <ul style="list-style-type: none"> • Set • Reset
Output Example 1		
<pre><OptIn xmlns=""><TimeStamp>6/29/2009 11:12:21 PM</TimeStamp> <Action>AutoWelcome</Action><Result>Set</Result> </OptIn></pre>		
Output Example 2		
<pre><OptIn xmlns=""> <TimeStamp>6/29/2009 11:12:21 PM</TimeStamp></pre>		



SOAP Web Management Services

```
<Action>AutoWelcome</Action><Result>Reset</Result>  
</OptIn>
```

SOAP Web Management Services

SendAutoWelcome

Function SendAutoWelcome(**ByVal** CellNumber **As String**) **As** XmlNode

This function sends the Auto Welcome message to the listed cell number. The carrier need not be supplied as it will be recovered from the OptIn information. Attempting to send an AutoWelcome message to a cell number not opted in for the campaign and keyword in the credentials will result in a failure.

Input Parameters		
Name	Data Type	Meaning
CellNumber	String	The number of the cell phone that is to have the message sent to it. The cell number must be exactly 10 digits for North American traffic.
Output Base Node:		OptIn
Sub Nodes		
Tag	Level-Type	Meaning
TimeStamp	2-Date	The date of the action
Action	2-String	The action that was performed. Always AutoWelcome Send .
CellNumber	2-String	The Cell phone number that the message was sent to.
Result	2-String	The result of the requested action. Possible results: <ul style="list-style-type: none"> • Sent • NotSent
Output Example 1		
<pre><OptIn xmlns=""><TimeStamp>6/30/2009 12:03:14 AM</TimeStamp><Action>AutoWelcome Send</Action><CellNumber>6345550144</CellNumber><Result>Sent</Result></OptIn></pre>		

SetOptInTimeout

Function SetOptInTimeout (ByVal CellNumber As String, ByVal Minutes As Integer) As XmlNode

The SetOptInTimeout function can be used when you have a need to automatically release an OptIn after a certain period of time. This can arise when using Virtual Dedicated Short Codes when a free dialog is needed but only for a limited period of time. You could also effect the release by using the ManageOptIn function and using your own timer. For users who cannot run timer functions on their own servers however, the SetOptInTimeout function is very useful. SetOptInTimeout will only work on an existing Optin which must have been previously created with an MO or a ManageOptIn call. When the OptIn is released, no message will be sent to the Cell Number that is being released.

Input Parameters		
Name	Data Type	Meaning
CellNumber	String	The number of the cell phone that is to have its OptIn released. The cell number must be exactly 10 digits for North American traffic.
Minutes	Integer	The number of minutes into the future that the OptIn is to be released. The timing of the OptIn release is not precise. It will never be less than the minutes specified but can be as many as 9 minutes later.
Output Base Node:		OptIn
Sub Nodes		
Tag	Level-Type	Meaning
TimeStamp	2-Date	The date of the action in Universal Time
Action	2-String	The action that was performed. The successful Action is TimeOut Set . If no action was performed
CellNumber	2-String	The Cell phone number that the Time Out was set on. The Short Code and Keyword of this cell phone number are derived from the credentials.
TimeOutAt	2-String	The time in Universal Time that the Optin will be released. The timing of the OptIn release is not precise. It will never be less than the time listed but can be as many as 9 minutes later.
Output Example 1		
<pre><OptIn xmlns=""><TimeStamp>11/18/2009 4:13:29 PM</TimeStamp> <Action>TimeOutSet</Action></pre>		



SOAP Web Management Services

```
<CellNumber>6305550156</CellNumber>  
<TimeOutAt>11/18/2009 4:18:29 PM</TimeOutAt></OptIn>
```

SOAP Web Management Services

ManageSubscription

`Function ManageSubscription(Action As String, CellNumber As String, Carrier As String, NoBill As Boolean, QuitReason as String) As XmlNode`

Manage a subscription service entry. If your subscription offering requires OptIn, you must set the OptIn separately before trying to add a subscription entry for them.

Input Parameters		
Name	Data Type	Meaning
Action	String	The Subscription action to be performed. Allowable case insensitive values are: add- create an entry for the subscription service delete – an entry in the subscription service query – Get the current subscription state increment – Increment the monthly download count
CellNumber	String	The number of the cell phone that is to be acted upon for the Subscription action. The cell number must be exactly 10 digits for North American traffic.
Carrier	String	The carrier of the Cell Number. If you are sure the carrier you have is accurate, you may enter it here. If you are not absolutely sure, leave it blank and the carrier will be looked up. A charge applies for the lookup. For billed subscription services, TextPower recommends leaving this parameter blank and doing the lookup. Important: If the carrier for this number is incorrect, the billing may not take place correctly!
NoBill	Boolean	Suppress the customer message that announces and bills the initial subscription offering or cancels an existing offering. This parameter should only be set to True for: <ol style="list-style-type: none"> 1. Subscription offerings that are free and do not bill anyway. 2. Doing maintenance on a subscriber who has had a problem 3. For test purposes to avoid racking up charges when developing a service. Setting this parameter to True will cause your billing on a customer to be erroneous, resulting in either no bills or in continuing to bill a

SOAP Web Management Services

		customer who should have been removed! Use with care!
QuitReason	String	Information for your use. You can record the reason for the quit here.
Output Base Node:		OptIn
Sub Nodes		
Tag	Level-Type	Meaning
TimeStamp	2-Date	The date of the information retrieval or action
CellNumber	2-String	The cell number on which the action was performed.
Status	2-String	<p>Subscription status of the cell number after the action was executed: Possible results: For command add:</p> <ul style="list-style-type: none"> • No OptIn • Already a Subscriber • Reinstated • Added • Excess Premium (This means that the charge for this subscription for this number would exceed carrier rules. Add not allowed) <p>For command reset:</p> <ul style="list-style-type: none"> • Short Code/Keyword Not Assigned • No Subscription Package Found • Not a subscriber • Deleted <p>For command query</p> <ul style="list-style-type: none"> • Short Code/Keyword Not Assigned • No Subscription Package Found • No OptIn • Not a Subscriber • Quit • Current Subscriber <p>For command increment</p> <ul style="list-style-type: none"> • Short Code/Keyword Not Assigned • No Subscription Package Found • Not a subscriber • Count Exhausted • Count Incremented
CurrentDownloadCount	Integer	The remaining download counts for this month.

SOAP Web Management Services

		The number of initial download counts is set on the setup page at the TextPower customer web site for your subscription.
SubscriptionID	Integer	The internal ID number of this subscription package.
JoinedFrom	2-String	A string indicating the source from which the user joined. Valid values are: MO meaning the user joined using an MO to a short code and keyword, API meaning from the classic ASP API call and SOAP meaning from a SOAP call.
QuitFrom	String	A code indicating how the subscriber quit the subscription list. Valid values are MO if the customer quit using a keyword to a short code, API if the user quit from an external source controlled by the application, SOAP if the user quit from the SOAP interface, or NMCC if the NMS Switch terminated the customer because of non-clearance of a billing charge.
QuitBecause	String	This field indicates the reason for the quit, if available. <ul style="list-style-type: none"> • The QuitBecause field is always empty if the QuitFrom source is MO-SP since no information can be collected. • If the QuitFrom source is NMCC, the actual billing error is listed. • If the QuitFrom source is API or SOAP, the reason as supplied by your API call is listed.
MessageStatus	String	The send status of the message sent for adds and for deletes. This is normally OK. Other values represent an error. If no message was sent, this tag does not appear.
Output Example 1 (add)		
<pre><Subscription xmlns=""> <TimeStamp>2/27/2008 9:46:19 AM</TimeStamp> <CellNumber>6345550144</CellNumber> <Status>Reinstated</Status> <CurrentDownloadCount>0</CurrentDownloadCount> <SubscriptionID>16</SubscriptionID> <JoinedFrom>SOAP</JoinedFrom> <QuitFrom>API</QuitFrom> <QuitBecause>T1</QuitBecause></pre>		

SOAP Web Management Services

```
<MessageStatus>OK</MessageStatus>
</Subscription>
```

Output Example 2 (query)

```
<Subscription xmlns="">
<TimeStamp>2/27/2008 9:51:43 AM</TimeStamp>
<CellNumber>6345550144</CellNumber>
<Status>Current Subscriber</Status>
<CurrentDownloadCount>-5</CurrentDownloadCount>
<SubscriptionID>16</SubscriptionID>
<JoinedFrom>API</JoinedFrom>
<QuitFrom>API</QuitFrom>
<QuitBecause>Business</QuitBecause>
</Subscription>
```

Output Example 3 (delete)

```
<Subscription xmlns="">
<TimeStamp>2/27/2008 9:55:18 AM</TimeStamp>
<CellNumber>6345550144</CellNumber>
<Status>Deleted</Status>
<CurrentDownloadCount>0</CurrentDownloadCount>
<SubscriptionID>16</SubscriptionID>
<JoinedFrom>API</JoinedFrom>
<QuitFrom>SOAP</QuitFrom>
<QuitBecause>Personal</QuitBecause>
<MessageStatus>OK</MessageStatus>
</Subscription>
```

Output Example 4 Attempt to add a subscription requiring OptIn with no OptIn set.

```
<Subscription xmlns="">
<TimeStamp>2/27/2008 10:02:58 AM</TimeStamp>
<CellNumber>6345550144</CellNumber>
<Status>No OptIn</Status>
<CurrentDownloadCount>0</CurrentDownloadCount>
<SubscriptionID>16</SubscriptionID>
<JoinedFrom>API</JoinedFrom>
<QuitFrom></QuitFrom>
<QuitBecause></QuitBecause>
</Subscription>
```