

---

Reference Guide for

# Spok Mobile with Pager Number

Spok, Inc.

## **Copyright**

Spok Mobile 4.0  
Release Date: February 16, 2015

Document Version 2.0

Copyright 2003-2015 Spok, Inc. All Rights Reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under one or more Program/Product License Agreements. The software may be used or copied only in accordance with the terms of those agreements. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any means electronic or mechanical, including photocopying and recording for any purpose without the written permission of Spok, Inc.

Spok, Inc.  
10400 Yellow Circle Drive  
Suite 100  
Eden Prairie, MN 55343  
USA

## **Trademarks**

Microsoft and Windows are trademarks of Microsoft Corporation. Other brands and their products are trademarks or registered trademarks of their respective holders and should be noted as such.

# Contents

- Spok Mobile with a Pager Number 1**
  - Spok Mobile with a Pager Number ..... 1
  - About This Guide ..... 1
    - Purpose ..... 1
    - Scope ..... 1
    - Intended Audience ..... 1
    - Guide Organization ..... 1
  
- WCTP 2**
  - WCTP ..... 2
    - Prerequisites ..... 2
  - Web.config File ..... 3
    - Log4net ..... 5
    - appSettings ..... 6
    - connectionStrings ..... 7
  
- Message Wipe API 8**
  - Message Wipe API ..... 8
    - Parameters ..... 8
    - Response ..... 9
  
- New Registration API 10**
  - New Registration API ..... 10
    - Send/Resend Email ..... 10
    - Response ..... 11
    - Registration Info ..... 12
    - Response ..... 13



# Spok Mobile with a Pager Number

---

## Spok Mobile with a Pager Number

The Spok Mobile with a Pager Number product is one configuration of Spok Mobile that allows devices to register with pager numbers instead of phone numbers, usernames, or email addresses.

---

## About This Guide

### Purpose

The purpose of this document is to provide internal Spok resources with the information that is needed to administer the Spok Mobile with a Pager Number product configuration.

### Scope

This document covers information on the Spok Mobile with a Pager Number configuration of the Spok Mobile product.

### Intended Audience

The intended audience for this document is internal Spok resources who are responsible for setting up and administering the Spok Mobile with a Pager Number configuration of the Spok Mobile product.

### Guide Organization

The following topics are covered in this document:

#### **WCTP**

“WCTP” on page 2 includes information on the protocol that needs to be set up and configured to enable the connection between the paging system and the Spok Mobile system.

#### **Message Wipe API**

“Message Wipe API” on page 8 includes information on the API that is used to wipe messages in the paging system and the Spok Mobile system.

#### **New Registration API**

“New Registration API” on page 10 includes information on the API that is used to handle new registration in the paging system and the Spok Mobile system configuration.

# WCTP

---

## WCTP

WCTP (Wireless Communications Transfer Protocol) is an XML-based communication protocol that allows two-way asynchronous communications between Spok Mobile and the paging system.

Before the gateway can be used to enable communication between the paging system and Spok Mobile, it must be configured in the WCTP web.config file. The web.config file is included below with the sections outlined within the file.

### Prerequisites

Before using the WCTP protocol, you must have the following things installed:

- IIS 7
- ASP.NET component enabled within IIS
- .NET Framework 4.0

---

# Web.config File

Within the web.config file, the following sections can be edited:

- Log4net
- appSettings
- connectionStrings

These sections are described below.

```
<?xml version="1.0"?>
<configuration>

  <configSections>
    <section name="providers" type="Amcom.Configuration.ConfigSection, Amcom.Clam" />
    <section name="log4net" type="log4net.Config.Log4NetConfigurationSectionHandler,
Log4net" />
  </configSections>

  <providers type="Amcom.Structure.SingletonProvidersSettings, Amcom.Common,
Culture=neutral" >
    <provider name="connectionManager" type="Amcom.Amc.Data.Access.Amc.ConnectionProvider,
Amcom.Amc.Data.Access.Amc, Version=1.0.0.0, Culture=neutral" />
    <provider name="logFactory" type="Amcom.Logging.Log4NetLogFactory, Amcom.Clam" />
  </providers>

  <log4net>
    <root>
      <level value="ALL" />
      <appender-ref ref="LogFileAppender" />
    </root>
    <appender name="LogFileAppender" type="log4net.Appender.RollingFileAppender, log4net">
      <param name="File" value="logs\wctp_log.txt" />
      <param name="AppendToFile" value="true" />
      <rollingStyle value="size" />
      <maxSizeRollBackups value="1" />
      <maximumFileSize value="5MB" />
      <staticLogFileName value="true" />
      <datePattern value="yyyyMMdd" />
      <layout type="log4net.Layout.PatternLayout">
        <conversionPattern value="%utcdate&#x9; [%thread]&#x9; %-5level&#x9; %message%newline"
/>
      </layout>
    </appender>
  </log4net>

  <system.web>
    <compilation debug="true" targetFramework="4.0" />
    <identity impersonate="false" />
  </system.web>

  <system.webServer>
    <modules runAllManagedModulesForAllRequests="false" />
    <handlers>
      <add name="wctpMessageHandler" path="wctp" verb="*"
type="Amcom.Amc.Services.Wctp.WctpMessageHandler" resourceType="unspecified"
precondition="integratedMode" />
    </handlers>
    <directoryBrowse enabled="false" />
  </system.webServer>

  <appSettings>
    <add key="wctpContactEmail" value="support@amcomsoft.com" />
    <add key="wctpContactPhoneNumber" value="18887977487" />
    <add key="wctpContactInternet" value="http://www.amcomsoftware.com/" />

    <add key="usmowctpServerUri" value="http://10.50.5.119/wctp" />

    <add key="MxppsSiteId" value="USMOTest" />
    <add key="MxppFromName" value="USA Mobility" />
    <add key="MxppConnectionRetryAttempts" value="5" />
    <add key="MxppChannelEndpointPort" value="20001" />
    <add key="MxppChannelEndpointAddress" value="localhost" />

    <add key="MxppMessageType.Reply" value="Amcom.Amc.Default" />
    <add key="MxppMessageType.NoReply" value="Amcom.Amc.NoReply" />
    <!--<add key="MxppMessageType.Reply" value="Amcom.Amc.CloseReply" />-->
    <!--<add key="MxppMessageType.NoReply" value="Amcom.Amc.CloseNoReply" />-->
```

```

<!--configure status message delivery and notification value below-->
<!--null value means that a message will not be forwarded to the wctp server-->
<add key="READ" value="READ" />
<add key="QUEUED" value="" />
<add key="FAILED" value="" />
<add key="IGNORED" value="READ" />
<add key="ENROUTE" value="" />
<add key="ACCEPTED" value="" />
<add key="DELIVERED" value="DELIVERED" />
<add key="ACKNOWLEDGED" value="READ" />

<!--mxpp queue configuration-->
<add key="mxpp.queue.retry.interval" value="00:00:30" />
<add key="mxpp.queue.max.age" value="2.00:00:00" />

<!--<add key="MxppsSiteId" value="USMOTest" />
<add key="MxppChannelEndpointPort" value="20221" />
<add key="MxppChannelEndpointAddress" value="localhost" />-->

<!--<add key="MxppsSiteId" value="USMOTest" />
<add key="MxppChannelEndpointPort" value="20001" />
<add key="MxppChannelEndpointAddress" value="10.50.9.144" />-->
</appsettings>

<connectionStrings>
  <add name="AMCModel"
connectionString="metadata=res://Amcom.Amc.Data.Access.Amc/AMCModel.csd|
res://Amcom.Amc.Data.Access.Amc/AMCModel.ssd|
res://Amcom.Amc.Data.Access.Amc/AMCModel.msl;provider=System.Data.SqlClient;provider
connection string="data source=localhost;initial
catalog=AmcomAmcPremiseCore;integrated security=False;User ID=sd;Password=thinksd;
multipleactiveresultsets=True;App=EntityFramework";
providerName="System.Data.EntityClient" />
  <!--<add name="AMCModel"
connectionString="metadata=res://Amcom.Amc.Data.Access.Amc/AMCModel.csd|
res://Amcom.Amc.Data.Access.Amc/AMCModel.ssd|
res://Amcom.Amc.Data.Access.Amc/AMCModel.msl;provider=System.Data.SqlClient;provider
connection string="data source=10.11.10.94;initial
catalog=AmcomAmcPremiseCore;integrated security=False;User ID=sd;Password=thinksd;
multipleactiveresultsets=True;App=EntityFramework";
providerName="System.Data.EntityClient" />-->
</connectionStrings>
</configuration>

```

## Log4net

<b>appender name</b>	The <b>appender name</b> field is not configurable.
<b>param name</b>	The <b>param name</b> field represents the name of the log file.
<b>Append to file</b>	<p>The <b>Append to file</b> field can be <b>True</b> or <b>False</b>.</p> <ul style="list-style-type: none"> <li>• <b>True:</b> You want to append the file. Every time you write to the file, a new file is generated.</li> <li>• <b>False:</b> You do not want to append the file. Every time you write to a new file, a new file is not generated.</li> </ul>
<b>rollingStyle value</b>	The <b>rollingStyle value</b> field represents what dictates if a new file is generated. For example, if you use “Size” as the value, a new file is generated based on the file size.
<b>maximumFileSize value</b>	The <b>maximumFileSize value</b> field dictates how large the file can be before a new file is generated. For example, if you use “5 MB” as a value, the file size grows until it is 5 MB and then a new file is generated after the file size reaches 5 MB. Please note that there is no maximum value that can be entered in this field.
<b>staticLogFileName value</b>	<p>The <b>staticLogFileName value</b> field can be <b>True</b> or <b>False</b>.</p> <ul style="list-style-type: none"> <li>• <b>True:</b> If <b>True</b> is used as the value, the name of the log file is static, or stays the same. Therefore, after the name of the log file is assigned, no additional changes to the name can be made.</li> <li>• <b>False:</b> If <b>False</b> is used as the value, the name of the log file is not static, or does not stay the same. Therefore, after the name of the log file is assigned, additional changes to the log file name can be made.</li> </ul>
<b>datePattern value</b>	<p>The <b>datePattern value</b> field dictates how you want the date that appends to the file name to display. For example, if you enter yyyMMdd as the value, that is how the date displays. The following date identifiers are identified with the following letters:</p> <ul style="list-style-type: none"> <li>• <b>Day:</b> “D”</li> <li>• <b>Month:</b> “M”</li> <li>• <b>Year:</b> “Y”</li> </ul>
<b>layout type</b>	The <b>layout type</b> field is not configurable.
<b>conversionPattern value</b>	The <b>conversionPattern value</b> field is not configurable.

## appSettings

<b>WctpContactEmail</b>	The <b>WctpContactEmail</b> field contains the email address that is contacted for matters regarding WCTP.
<b>WctpContactPhoneNumber</b>	The <b>WctpContactPhoneNumber</b> field contains the phone number that is contacted for matters regarding WCTP.
<b>WctpContactInternet</b>	The <b>WctpContactInternet</b> field contains the URL for the web site that can be accessed for matters regarding WCTP.
<b>UsmoWctpServerUri</b>	The <b>UsmoWctpServerUri</b> field contains the uniform resource identifier information for the Spok WCTP server.
<b>MxppSiteId</b>	The <b>MxppSiteId</b> field contains the site ID that is assigned to the location where the MXPP gateway is located.
<b>MxppFromName</b>	The <b>MxppFromName</b> field contains the name for the location where the MXPP gateway is located.
<b>MxppConnectionRetryAttempts</b>	The <b>MxppConnectionRetryAttempts</b> field dictates how many times the system attempts to connect to the MXPP gateway. For example, if you enter “5” into the field, the system tries to establish a connection five times before stopping.
<b>MxppChannelEndpointPort</b>	The <b>MxppChannelEndpointPort</b> field contains the port number through which the WCTP protocol connects to the MXPP protocol.
<b>MxppMessageType</b>	The <b>MxppMessageType</b> field allows you to identify the MXPP message types. For example, you can create Reply and No Reply message templates.
<b>add key</b>	<p>Manages the message statuses. Please note that a null value means that a message is not forwarded to the WCTP server.</p> <ul style="list-style-type: none"> <li>• <b>Read:</b> A message is considered read if the message was read by the message receiver on his or her device.</li> <li>• <b>Queued:</b> A message is considered queued if the message is currently in the queue and is waiting to be sent to the message receiver’s device.</li> <li>• <b>Failed:</b> A message is considered failed if the message could not be successfully sent to the message receiver.</li> <li>• <b>Ignored:</b> A message is considered ignored if the message receiver ignored the message.</li> <li>• <b>Enroute:</b> A message is considered enroute if the message is currently traveling to the message receiver’s device.</li> <li>• <b>Accepted:</b> A message is considered accepted if the system has confirmed that it can successfully send the message to the message receiver’s device.</li> <li>• <b>Delivered:</b> A message is considered delivered if the message has successfully reached the message receiver’s device, but the receiver has not yet chosen a response option.</li> <li>• <b>Acknowledged:</b> A message is considered acknowledge if the message receiver viewed and acknowledged the message on his or her device.</li> </ul>

<b>Mxpp.queue.retry.interval</b>	The <b>Mxpp.queue.retry.interval</b> field dictates the duration of time that the system waits between retrying to establish a connection. Please note that this field is measured in the following way: HH:MM:SS.
<b>Mxpp.queue.max.age</b>	The <b>Mxpp.queue.max.age</b> field dictates the maximum number of days, hours, minutes, and seconds that messages are held in the message queue before being deleted. Please note that this field displays in the following format: DD.HH:MM:SS.
<b>MxppSiteId</b>	The <b>MxppSiteId</b> field contains the site ID value that is assigned to the site where the MXPP gateway is located.
<b>MxppChannelEndpointAddress</b>	The <b>MxppChannelEndpointAddress</b> field contains the address for the MXPP channel.

## connectionStrings

<b>AMCModel</b>	The <b>AMCModel</b> field contains the information for the location of the Spok Mobile database (Premise Core).
-----------------	---

# Message Wipe API

---

## Message Wipe API

The API that is used to wipe message for devices is the following:

GET

`http://<SERVER>/AMC/Message/Message.svc/RemoveAll?siteId='<SITE_ID>'&localDeviceId='<LOCAL_DEVICE_ID>'&callbackUrl='<CALLBACK_URL>' HTTP/1.1`

### Parameters

<b>SITE_ID</b>	Enclose with single quotes. For example: 'MedicalSite1'. This value is required. It represents the SITE_ID of the registration for which the messages are wiped.
<b>LOCAL_DEVICE_ID</b>	Enclose with single quotes. For example: 'K198798'. This value is required. It represents the LOCAL_DEVICE_ID from the host system or the Registrations table.
<b>CALLBACK_URL</b>	Enclose with single quotes. For example: 'http://www.somesite.com/Callback'. This value is optional. It is called upon the completion of the wipe operation.

### Event Flow

1. The procedure saves a new WipeMessages command in the Commands table.
2. A push notification is sent to the device using the device NotificationAddress to notify it that it has a new command.
3. Using the ClientMessaging service in Spok Mobile Services, the status of all the messages for this device are set to WIPED.
4. When the device receives the push notification, it pulls the new WipeMessages command and wipes its messages locally.

## Response

HTTP/1.1 200 OK

```
<RemoveAllResponse xmlns="http://tempuri.org/">  
<RemoveAllResult>true</RemoveAllResult>  
</RemoveAllResponse>
```

The call returns “false” in “RemoveAllResult” in one of the following cases:

- No valid registration is found for the given SITE\_ID and LOCAL\_DEVICE\_ID.
- The OutputGateway column of the found registration record is not populated and cannot be populated from an existing message sent to this registration.
- A valid record for “command.service.url” does not exist in the CONFIGURATION\_INFO table.
- An associated device record in the Devices table does not exist for the found registration.
- The request that changes the status of the messages to ‘WIPED’ fails for some reason, or the call returns “true” in “RemoveAllResult”.

# New Registration API

---

## New Registration API

### Send/Resend Email

```
POST http://<SERVER>:8090/ClientRegistration HTTP/1.1
```

```
Host: <SERVER>:8090
```

```
Content-Type: text/xml; charset=utf-8
```

```
Content-Length: length
```

```
SOAPAction: "http://amcomsoft.com/ResendEmail"
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
```

```
  <s:Header>
```

```
    <Action s:mustUnderstand="1"
xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://amcomsoft.com/ResendEmail</Action>
```

```
  </s:Header>
```

```
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
    <ResendEmail xmlns="http://amcomsoft.com/" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
```

```
      <SiteID>string</SiteID>
```

```
      <LocalDeviceID>string</LocalDeviceID>
```

```
    </ResendEmail>
```

```
  </s:Body>
```

```
</s:Envelope>
```

## Response

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: [length](#)

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <ResendEmailResponse xmlns="http://amcomsoft.com">
      <ResendEmailResult>string</AddRegistrationResult>
    </ResendEmailResponse>
  </s:Body>
</s:Envelope>
```

Result	Situation
Sent	An email is sent.
No email address	An email address is not available.

## Registration Info

The API used to get device and registration information about a device for a specific site.

```
POST http://<SERVER>:8090/ClientRegistration HTTP/1.1
```

```
Host: <SERVER>:8090
```

```
Content-Type: text/xml; charset=utf-8
```

```
Content-Length: length
```

```
SOAPAction: "http://amcomsoft.com/RegistrationInfo "
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
```

```
  <s:Header>
```

```
    <Action s:mustUnderstand="1"
xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://amcomsoft.com/RegistrationInfo</Action>
```

```
  </s:Header>
```

```
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
    <RegistrationInfo xmlns="http://amcomsoft.com/">
```

```
      <SiteID>string</SiteID>
```

```
      <LocalDeviceID>string</LocalDeviceID>
```

```
    </RegistrationInfo>
```

```
  </s:Body>
```

```
</s:Envelope>
```

## Response

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: [length](#)

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <RegistrationInfoResponse xmlns="http://amcomsoft.com">
      <RegistrationInfoResult>
        <siteId>string</siteId>
        <registrationId>long</registrationId>
        <localDeviceId>string</localDeviceId>
        <telephoneNumber>string</telephoneNumber>
        <emailAddress>string</emailAddress>
        <carrier>string</carrier>
        <deviceOSInfo> string</deviceISInfo>
        <status>string</status>
        <gateway>string</gateway>
        <clientversion>string</clientversion>
        <notificationAddress>string</ notificationAddress >
        <vendorDeviceID>string</ vendorDeviceID >
        <messageVersion>string</ messageVersion >
        <deviceConfirmed>string</ deviceConfirmed >
        <password>string</ password>
        <currentEncryptionMethod>string</ currentEncryptionMethod >
        <currentEncryptionKey>string</ currentEncryptionKey >
        <PendingEncryptionMethod >string</ PendingEncryptionMethod >
        <PendingEncryptionKey >string</ PendingEncryptionKey >
        <port>string</ port>
        <pendingport>string</ pendingport>
      </RegistrationInfoResult>
    </RegistrationInfoResponse>
  </s:Body>
```

</s:Envelope>

<b>Status</b>	<b>Situation</b>
<b>NEW</b>	New registration.
<b>UPDATED</b>	Updated by device and not received by the host system (Spok Mobile devices registered with a pager number are perpetually in this state because device confirmation is not used).
<b>CONFIRMED</b>	Device and host system in are in sync.
<b>ACTIVE</b>	Once the host system has called CheckForConfirmations, the registration status becomes active, letting Spok Mobile know that the host system received the information.