IntelliMessage System Description and Message Flow

Version 1.1
October 24, 2012

American Messaging Services, LLC.
1720 Lakepointe Drive
Lewisville, Texas  75057

CONFIDENTIALITY NOTICE

This document contains information which is confidential and proprietary to American Messaging Systems, LLC. Unauthorized use, disclosure or reproduction of information contained in this document is strictly prohibited.
# IntelliMessage System Description and Message Flow

## TABLE OF CONTENTS

1. Document History ........................................................................................................... 3

2. Introduction .................................................................................................................. 4

3. Message Flow Detail ..................................................................................................... 6

   3.1 IntelliMessage Desktop - Message Flow .................................................................. 6

   3.2 IntelliMessage Desktop Message Flow .................................................................... 7

   3.3 IntelliMessage Desktop Security ............................................................................. 7

   3.4 IntelliMessage Mobile – Android, iOS (iPhone) ..................................................... 7

   3.5 IntelliMessage Mobile Security .............................................................................. 8

   3.6 IntelliMessage Mobile Message Flow .................................................................... 8

   3.7 Message Flow Diagram ........................................................................................... 9

4. System Functionality ..................................................................................................... 10

   4.1 IntelliMessage Desktop Processing Messages and Errors ..................................... 10

      4.1.1 Check Message ............................................................................................... 10

      4.1.2 Get Message .................................................................................................. 10

5. IntelliMessage Software Block Diagram ...................................................................... 12
## 1. Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>2-8-2012</td>
<td>John Nagel / Peter Barnett</td>
<td>First Draft</td>
</tr>
<tr>
<td>0.8</td>
<td>5-11-2012</td>
<td>John Nagel</td>
<td>Minor updates, grammar, etc.</td>
</tr>
<tr>
<td>1.1</td>
<td>10-24-12</td>
<td>Peter Barnett / John Nagel</td>
<td>Changes for Release to Market</td>
</tr>
</tbody>
</table>


2. Introduction

The IntelliMessage system is an extension to American Messaging’s nationwide paging system. Its main function is to deliver messages that were delivered to a pager, to other devices such as smartphones and desktop PCs.

The IntelliMessage system has 2 main client software packages, IntelliMessage Mobile and IntelliMessage Desktop. IntelliMessage Mobile consists of a client software package for both Android and Apple iPhone smartphones. IntelliMessage Desktop consists of a client software package for Microsoft’s Windows and Apple’s Mac PCs.

Below is a top level diagram showing the major components of the IntelliMessage system which includes the various types of devices that the system communicates with, American Messaging’s paging system and the IntelliMessage system servers.

IntelliMessage Desktop is a messaging application (Client) that runs on your computer either Windows or Apple Mac. Much like instant messaging the IntelliMessage system allows you to send and receive messages immediately. Incoming messages will automatically pop up on your computer display as soon as they arrive and allow you to immediately reply with the press of a button. You may also create quick messages to send to any email address.

IntelliMessage Mobile is very similar to IntelliMessage Desktop except it runs on an Android and Apple iOS (iPhone) based smartphones.
IntelliMessage System Description and Message Flow

IntelliMessage Desktop and Mobile provide the following features and functionality:

- Instantly Receive Messages that were sent to your Pager
- Send Messages
- Reply to Received Messages (must be from a repliable address)
- Track Message History
- Ability to Check for Messages
- Perform Administrative Functions
- Provides Quick Access to Web and Other Information

The following sections will discuss how the messages flow through the system.
3. Message Flow Detail

3.1 IntelliMessage Desktop - Message Flow

The following text and the diagram shown in section 3.7 describe how the IntelliMessage system interacts with the IntelliMessage desktop application.

The AMSPush server handles what we call “long-polling” for the Desktop client applications via https protocol. The desktop application sends a web service request to the AMSPush server to check for pending messages and waits for a reply.

See Wikipedia at http://en.wikipedia.org/wiki/Long_polling#Long_polling and it explains; “Long polling is a variation of the traditional polling technique and allows emulation of an information push from a server to a client. With long polling, the client requests information from the server in a similar way to a normal poll. However, if the server does not have any information available for the client, instead of sending an empty response, the server holds the request and waits for some information to be available. Once the information becomes available (or after a suitable timeout), a complete response is sent to the client. The client will normally then immediately re-request information from the server, so that the server will almost always have an available waiting request that it can use to deliver data in response to an event. In a web/AJAX context, long polling is also known as Comet programming.

Long polling itself not a push technology, but can be used under circumstances where a real push is not possible or required. Long polling is the fastest way a remote device can see if there is a “New” message available. And normally that would indicate to the client that they need to go someplace else to actually retrieve the messages. Laptop battery life will be minimally affected by this type of service. No different than refreshing a web page every 60 seconds.

The AMSRest provides web site and data retrieval via https. Message history, message retrieval, administrative web pages, and downloading of updates to the PC Desktop application also go through the AMSRest server Internet Information Services (IIS) service.

The web service process on the server polls the database at short intervals, checking for new messages for the user account. If any are found, it replies to the desktop application that new messages are available. After one minute, if no messages have arrived, it replies that there are none, and the client app starts another long-poll by starting a new web service request. If new messages exist, the client app makes a separate web service call to the AMSRest server to retrieve the message(s).

The AMSMail server handles incoming messages, parsing them and writing the data into an inbound message queue data on the SQL server, AMS Msg. No outgoing messages go through this server.

The AMSMsg server hosts the SQL Server database for the Rest, Push and Mail servers. It sends all outgoing messages. Only one AMSMsg server will be used for production, with the second server functioning as a warm-standby, its database being always replicated from the first. It’s likely we’ll directly connect the live and standby servers via a cross-over CAT5 cable for this data replication, rather than sending it through the switch.
IntelliMessage System Description and Message Flow

3.2 IntelliMessage Desktop Message Flow

Refer to the drawing in section 3.7. The numbered colors below correspond to the colored lines with arrows in the drawing.

(1) As an example message is sent from the AMS paging system or from a user containing a message. The email goes through Spam Filter ISP and Spam Assassin servers to the AMSMail server. AMSMail parses the message and writes the contents to an incoming message table on AMSMsg.

(2) The AMSMsg server removes the message from the incoming table, processes it, and writes it to the received messages database.

(4) If a desktop client is being used, a long-poll web service process on AMSPush and detects the new message in the received messages database and notifies the desktop application that a new message has arrived.

(5) The desktop / laptop device calls a web service on AMSRest, requesting new messages.

(7) For each device that retrieves the message, AMSMsg stores the response and if programmed, sends an email to the email address that sent the original message, notifying them that the message has been received by the device.

3.3 IntelliMessage Desktop Security

All communication after the initial download of the client is done via HTTPS. The initial server that we download the client from is token authenticated between the client applications and the host site. All of the code has been written and / or authenticated by American Messaging Personnel. American Messaging owns all of the code associated with the server and client applications.

3.4 IntelliMessage Mobile – Android, iOS (iPhone)

The primary method for receiving messages for the IntelliMessage Mobile – Android is the same long polling process as the IntelliMessage desktop. However, unlike the IntelliMessage desktop applications, mobile devices may fade in and out of coverage during a long poll process. In order to insure that messages are delivered timely the push process is also used and is provided as a backup to the polling process. The primary method for receiving messages for the IntelliMessage Mobile - iOS is the Apple push process originating from the IntelliMessage servers.

The mobile applications that are using long polling start a long poll process approximately every 2 minutes. However, if a message has been waiting for delivery for 3 minutes, the server will perform a push process to the device. The mobile device may be out of coverage, turned off, or the battery may be low and the device has restricted certain functionality as part of its “Low Battery” settings. Polling and the push timing parameters can be set at the user group level in the server. Short polling times means faster message delivery, but also requires more battery capacity/use, hence a shorter battery life. Also, different mobile operating systems and manufacturer “skin layers” on Android devices will handle open applications differently as they try to maximize either the battery life or the use of memory on the device.
IntelliMessage System Description and Message Flow

The AMSMail server handles incoming messages, parsing the message and writing the data into an inbound message queue on the AMSMsg SQL server. When the AMSMsg server detects that a new message is queued for a mobile device and that device has not polled for 3 minutes, it sends a message alert (push) to, depending on the type of device, either the Apple Push Notification Services for iOS based iPhones (http://en.wikipedia.org/wiki/Apple_Push_Notification_Service) or the Android Cloud to Device Messaging (C2DM) for Android phones (http://code.google.com/android/c2dm/).

The push alert to the device triggers the application on the device to start a message retrieval process, which then retrieves the message from the AMSRest server via https. Message history, message retrieval, and administrative web pages, also go through the AMSRest server Internet Information Services (IIS) service via https.

The AMSMsg server hosts the SQL Server database for the Rest, Push and Mail servers. It sends all outgoing messages. Only one AMSMsg server will be used for production, with the second server functioning as a warm-standby, its database being always replicated from the first.

3.5 IntelliMessage Mobile Security

All communication after the initial download of the mobile application is done via HTTPS. The initial server that we download the client from is token authenticated between the client applications and the host site. All of the code has been written and / or authenticated by American Messaging Personnel. American Messaging owns all of the code associated with the server and client applications. Mobile devices will get their applications through the appropriate OS market.

3.6 IntelliMessage Mobile Message Flow

(1) As an example, a message is sent from the AMS paging system or from a user containing a message.

The message goes through SpamFilter ISP and Spam Assassin servers to the AMSMail server. AMSMail server parses the message and writes the contents to an incoming message table on AMSMsg.

(2) AMSMsg removes the message from the incoming table, processes it, and writes it to the received messages database.

(3) For a mobile message, AMSMsg notifies either the Apple Push Notification Services for iOS based iPhones or the Android Cloud to Device Messaging (C2DM) for Android phones that a “Message Notification Push” needs to be sent to the appropriate mobile device.

(6) The push notification on the device triggers the application to call a web service on AMSRest, requesting new messages. If the device does not get a push for 3 minutes (programmable), the device will start to poll the AMSRest server requesting new messages. The polling will stop and the timer will reset once a new push message is received.
IntelliMessage System Description and Message Flow

For each device that retrieves the message, AMSMsg stores the response and if programmed, will send an email to the email address that sent the original message, notifying them that the message has been received by the device.

3.7 Message Flow Diagram

The drawing shown below shows the message flow detail that is described in the above paragraphs.
4. System Functionality

4.1 IntelliMessage Desktop Processing Messages and Errors

4.1.1 Check Message

1. Long Polling Process starts when application is run, makes web request and waits for server response. Processes server response or generates error response.
   a. Server response.
      i. Message found.
         1. Web request for message information and display message.
      ii. No Message.
         1. Long Polling continues.
   b. Error response.
      i. Check to see if issue is because internet access is lost.
         1. If so popup toast message regarding lost of access. Long Polling continues to try and re-establish connection to servers.
      ii. Internet access available.
         1. Pop up message polling offline.
         2. Back up polling via timer started. Checks every 15 seconds for message.
         3. Long Polling continues to try and re-establish connection to servers. Once successful, back up process terminated.
      iii. Internet access not available.
         1. Pop up toast message “Internet access is down. Application offline until internet is back up”.

4.1.2 Get Message

1. Web request for message information and display message.
2. Stop other timers and processes that could interrupt message processing.
3. Error response.
   a. Internet access available.
      i. If so Display pop up message that message could not be processed.
      ii. Application will request message information again is internet access available.
   b. Internet access not available.
      i. Pop up toast message “Internet access is down. Application offline until internet is back up”.
4. Display message.
   a. If subject contains /RR then Message Received is sent to message sender.
   b. User Responds.
      i. Message marked as read.
      ii. Sender button.
         1. Response sent and check for new messages.
      iii. System Generated ### Message Read and Create Reply Request.
         1. Message Read.
IntelliMessage System Description and Message Flow

a. Response sent and check for new messages. No new messages timers and background processes restarted.

2. Create Reply selected.
   a. Application opens Send Message Screen, after message sent or screen closed, return to Get Message screen and check for new messages. No new messages timers and background processes restarted.

c. User does not respond.
   i. Start timer to check if current message is responded too.
      1. Wait 30 seconds and issue popup “You have an open Message. Please respond” and beep warning tone.
      2. Repeat 5 times and then stop.
      3. Application waits for user to respond to message. If application closed or connection lost message will be displayed again next time application connected to servers.
5. IntelliMessage Software Block Diagram