



MyMobileAPI
mymobileapi.com

Windows Service Overview

TABLE OF CONTENTS

Overview	3
Architecture	4
Configuration	4



OVERVIEW

The Windows service is a 2 part application consisting of the following parts:

- Windows service (written in Microsoft .Net 2)
- SQL Server database

In order to utilize the service, Microsoft .Net 2 runtime and Microsoft SQL server (MS SQL) must be installed. The service simplifies the transport layer between an application (local environment) and the SMS Gateway. The user can simply write data to a QUEUE table and have the service take care of the communication layer. It also caters for all updates of delivery receipts / replies.

The service was written for multi user accounts. Each user account can have different properties set that include:

- Active (if false it won't send for that specific user)
- Start and End hour (only send between the following hours)
- Poll interval on the QUEUE (the more aggressive the more the machine will work)
- Interval to fetch receipts and replies (the more aggressive the more bandwidth is consumed)

The service allows the following when sending:

- Queue messages into the future
- Assign priority to each message (the higher the priority the faster the service will process it)
- Send SMS, WAP push and VCARD messages

ARCHITECTURE

The Windows service was designed using a multithreaded Object Orientated architecture to simplify any future additions and to allow clients to customize the service to their specific requirements.

The threads work seamlessly together to create a non-blocking environment as to improve performance and message throughput. The throughput however is determined by the speed of the internet connection being utilized.

CONFIGURATION

The config file accompanying the service caters for specific database and network connection settings. This includes the connection string to the database (which can be on a different server) and the connection settings for your internet connection. This is only relevant if a firewall / proxy server is being utilized in your environment.

All settings are controlled via the database and allows sending of data to be paused for a user mid-stream. The paused messages can then be removed from the QUEUE or delayed for a future date