Lone Worker Developer Guide

SONIM CONFIDENTIAL

Sonim Technologies, Inc.
1875 South Grant Street, #620 Floor
San Mateo, CA 94402, USA
www.sonimtech.com
+1 (650) 378 8100

REVISION HISTORY

Revision	Date	Author	Reviewers	Comments
0.1 – 1.06	08-06-2009	Gomu	Adithya, Biswajit, Sidda, Dora, Sanjay, Will, Harish, Tony	Initial version
PB10	05-06-2010	Gomu		Keypad capture and automatic mode change
PB11	05-27-2010	Gomu		Removed getconfig all in changeconfig, Added CP1.1 details for setting APN, Support for specifying port number, New parameter to restart phone
PB12	05-28-2010	Gomu		Review comments on best practices incorporated
PB13	06-02-2010	Harish		Review comments on best practices incorporated
PB14	06-07-2010	Gomu		Review comments from partner incorporated, new parameters IncludeTimeZone and UseShortenedMessages

CONTENTS

1 Introduction

The Sonim XP3 Sentinel is designed for Lone Worker market with features such as, periodic sending of GPS location and other phone health status to the server, detection of tilt and non-movement of the phone and sending multiple alert levels to the server, over the air configuration and control of the client from the server and much more. Furthermore, it has the same highly rugged design as the XP3 Quest to sustain even the toughest work environments. It conforms to IP67, MIL-810F standards and Sonim's RPS test specification confirming to tests such as conform to 2m (6ft) drop on hard concrete from any angle and high capacity battery for long operations.

The native Lone Worker Client application (client) can be integrated with a Lone Worker Server application (server). This document gives the details of the behavior of the client and its interaction with the server. It is designed to support a variety of lone worker solutions offered by existing lone worker service providers, monitoring centers, system integrators and value added resellers. Besides the lone worker service provider channel, XP3 Sentinel is also expected to find application with enterprises and government organizations that deploy commercial lone worker system software and operate their own monitoring capabilities.

Sonim provides technical support to partners to understand the Lone Worker client and its interface with the server. Sonim has a reference implementation of the server, which can be made available to partners to understand the protocol better.

This developer guide is for Sonim partners of Lone Worker server systems and gives details of the behavior of the XP3 Sentinel Lone Worker functionality and its interaction with the server system.

2 Acronyms

General Terms

Acronym	Definition
Client	Lone Worker client
Server	Lone Worker server
ERC	Emergency response center
Long beep	Message that is played when there is an alert to the user
Short beep	Message that is played when the user has been informed and waiting for further action

Lone Worker Modes

Mode	Definition
Disabled mode	No functionality is enabled
Simple mode	Only manual alarm and the various checks like GSM, GPS, etc are enabled

Tracking mode	In addition to Simple mode, the location is sent to the server periodically
Man down mode	In addition to Tracking mode, tilt detection and idle detection are active
Alarm mode	The user is in distress

Incident Reporting Modes

Mode	**Definition**
Local warning	A local alert is shown to the user. If the user does not acknowledge it, it is closed automatically after some time.
Local alarm	A local alert is shown to the user. The user has to acknowledge it.
Remote alarm	A local alert is shown to the user. If the user does not cancel it, alarm is sent to the server.
Remote alarm with call	A GSM call to the ERC is also established with the behavior of *remote alarm.*
Silent call	Alarm is sent to the server and a GSM call is made to the ERC. All local audible indications are disabled till connecting to the ERC.
Remote report	Report is sent to the server with guaranteed delivery.
Remote info	Report is sent to the server with guaranteed delivery. Also, a local alert is shown to the user. If the user does not acknowledge it, it is closed automatically after some time.
No report	Ignore the incident.

Transport Modes

Mode	**Definition**
Try once	Try to deliver message once. If not possible, drop it.
Try later	If it is not possible to deliver now, try later. If queue is full, drop it.
Send now or later	If it is not possible to deliver now, try later.
Send now	If it is not possible to deliver now, try the backup transport (like SMS).

More details about these are in tables in the document.

3 High-level Functionality Summary

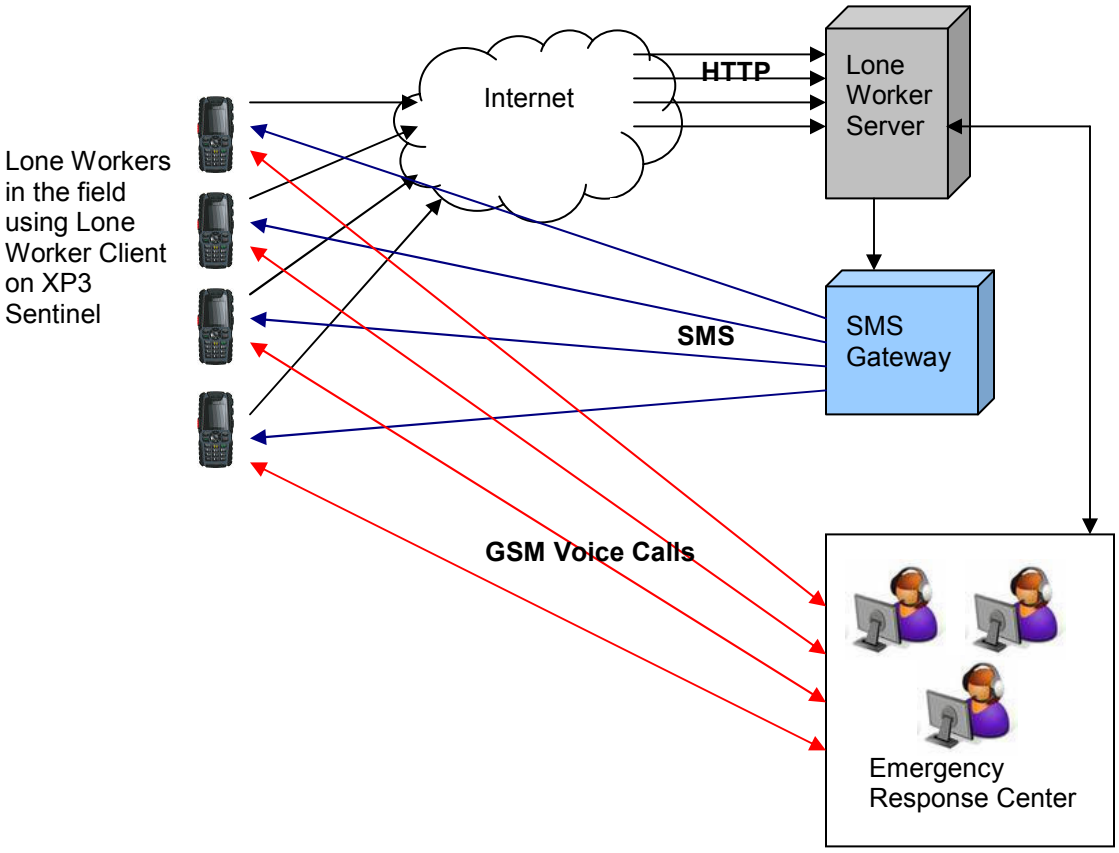
- The Lone Worker application runs in the native environment and works in conjunction the J2ME environment with JAVA API support. This allows for flexibility third party development

customization of Lone Worker application based its special needs, such as user login before any/some lone worker functionality will be initiated or give user access to modify some of the configuration parameters.

- The Lone Worker application have several trigger points like timer, self test failure, GPS out of coverage, GPRS out of coverage, low battery, call from known emergency numbers, man down, manual alarm, entering or exiting high risk zone, etc.
- Each trigger point is configurable with respect to enable/disable and thresholds.
- The Lone Worker application has several actions that can be taken on the trigger points.
- Upon some emergencies the Lone Worker application can connect the user via GSM voice call to an Emergency Response Center (ERC) which is configured in the client.
- The Lone Worker application sends GPS position data either via GPRS/EDGE or SMS if tracking has been enabled.
- The Lone Worker application can be configured remotely through OTA SMS.
- Three buttons – red, amber and green are provided on the phone. Long press of the red button should raise an alarm. Amber and green buttons are mapped to launch Java applications that can change the mode of the Lone Worker application.
- Audio alerts should be used to alert the user about different situations. All the audio files should be configurable.

4 Solution Architecture

The complete Lone Worker System consists of the XP3 Sentinel, a partner server and an Emergency Response Center (ERC).



Sonim provides the Lone Worker Client that is tightly integrated with the phone. The behavior of the Client and the interactions with the rest of the system are well documented in this Developer Guide.

The partner develops the Lone Worker Server. Sonim provides technical support and some reference code.

The service provider uses the combined solution provided by Sonim and the partner to provide the service to end user. This may be the same as the partner.

The end user is a company who has several workers in the field. This may be the same as the service provider.

The XP3 Sentinel and the Lone Worker Client are designed to conform to the different standards defined for Lone Worker Solutions.

5 Use Cases (Informative)

This section describes some of the many use case scenarios for how the XP3 Sentinel can be used.

5.1 Lone worker system

Assume a company which has a number of its workers working alone at different locations and also traveling. The company has integrated Sonim Lone Worker client with its web based configuration and monitoring servers. There is an Emergency Response Center (ERC) which handles emergency calls from and to the phones. Each lone worker is given a phone with Sonim Lone Worker client on them.

5.2 System closure

The secret key used to identify this entire system and encrypt all messages in this system is configured on the server and each of the phones manually. This ensures that no intrusion is possible into the system. This does not allow a Sonim Lone Worker client to work with the server belonging to a different deployment of the same system.

5.3 General provisioning and configuration

The company has several options to distribute the basic configuration.

1. The basic configuration is put into SD cards and put into the phones.
2. The basic configuration is put into the phones using Bluetooth.
3. A java based configuration tool is used to create a configuration, which is loaded into the Lone Worker client.
4. The basic configuration is put into the phone using the web based configuration system using OTA SMS and HTTP.

If the unique ID and username for each phone is not put in as a part of the SD card configuration or the Bluetooth configuration, it is pushed from the server to each phone by SMS.

Any configuration tuning for each phone is done from the server. Anytime, any parameter can be changed from the server.

Note: Whenever the configuration is changed, the user should restart the application for the configuration to take effect and test it well before using it in the field.

5.4 Java based configuration tool

A user of the lone worker client is allowed to change some parameters on his phone. His phone has a java application which can be used to change the parameters and create a configuration file. The updated configuration will take effect when the Lone Worker client is restarted. This Java application can be customized for different Lone Worker deployments. The splash screen, the colors and the list of configurable parameters can be customized.

5.5 Fixed workers

Some workers in the system are not moving from one place to another. For these people, the exact location need not be known. So, instead of using GPS as the location source, CellID is used. GPS consumes more battery.

5.6 Traveling workers

Some workers are often on the move. But they are in general not in any hazardous situation. These people need to be tracked always. The minimum lone worker mode is set to tracking for these users.

5.7 Patrolling workers

Some workers are on duty to keep patrolling. They should be always on their feet. They do night shifts also. So we should make sure that they are awake. The minimum lone worker mode is set to man down for these users. Idle detection and tilt detection are enabled on these phones. The awake check system is enabled and configured to check if the person is awake every 30 minutes.

5.8 Changing the mode

The user can set the client to one of the several modes: disabled, simple, tracking and man down. He can go to Menu > Applications and select Lone worker. He is shown the current mode. Here he can change the mode by using the up and down navigation keys, and selecting OK. The server and Java applications can change the mode too. Long press of the amber button usually sets the mode to tracking and long press of the green button usually sets the mode to simple.

5.9 Lone worker falls asleep

The lone worker who works in the night shift in a non-threatening situation has his phone in tracking mode. He has his awake check configured to check every 30 minutes. Every 30 minutes, it starts ringing, which he has to stop. If he falls asleep and wakes up with in 10 seconds of the alarm ringing, he stops the alarm. If he does not stop the alarm, the ERC is notified. The ERC changes the lone worker mode to

"alarm" state and calls the phone. The person wakes up and attends the call. The officer changes the lone worker mode back to tracking mode.

5.10 Lone worker is incapacitated

If the lone worker whose idle detection is enabled does not keep walking for 10 seconds, his phone starts ringing. If he does not stop the alarm, the phone goes into alarm mode and calls the ERC. The ERC is able to listen to the sounds around. When the ERC person makes sure that there is nothing suspicious, he tries to talk to the worker. He does not respond. The ERC dispatches an emergency rescue team to the location of the worker as got from the phone.

5.11 Lone worker falls down

If the lone worker whose tilt detection is enabled falls down and is not able to get up within 10 seconds, his phone starts ringing. If he does not stop the alarm, the phone goes into alarm mode and calls the ERC. Now the officer is able to get back and does not need help. He conveys the same to ERC and changes the lone worker mode back to man down mode.

5.12 Lone worker indoors

A lone worker who works in a hazardous indoor environment like a warehouse has idle and tilt detection enabled but GPS disabled to conserve battery.

5.13 Traveling worker in poor network coverage

A traveling worker's phone is configured to send his location to the headquarters every 5 minutes. He passes through a place where there is poor network coverage where GPRS does not work but GSM works. The location messages are queued to send to the headquarters later. When he gets into a place where there is GPRS coverage, all the queued messages are sent.

5.14 Traveling worker in trouble

A traveling worker faces an emergency situation. He presses and holds the alarm button (Red key). The alarm is raised to the headquarters by GPRS and a stealth call is placed to the ERC. The headquarters has sent the information to the ERC. The ERC listens to the sounds around and takes appropriate action.

5.15 Traveling worker in trouble in poor network coverage

A traveling worker passes through a place where there is poor network coverage where GPRS does not work but GSM works. At this time, he faces an emergency situation. He presses and holds the alarm button (Red key). The alarm is raised to the headquarters by SMS and a stealth call is placed to the ERC. The headquarters has sent the information in the SMS to the ERC. The ERC listens to the sounds around and takes appropriate action.

5.16 Traveling worker in trouble in no network coverage

A traveling worker passes through a place where there is no network coverage even GSM does not work. When he enters such a place, the client raises a local alarm to the user to indicate to him that there is no network coverage. The user acknowledges the alarm. During this time, he faces an emergency situation. He presses and holds the alarm button (Red key). The phone goes into alarm mode. But because there is no network coverage, it cannot contact the server or the ERC. However, it collects the GPS coordinates and other system parameters and keeps queuing them. It keeps trying periodically if it can send the alarm to the server and call the ERC. When the user gets back into network all the queued messages are sent to the server and a GSM call is placed to the ERC.

5.17 Audio routing

If the lone worker has no Bluetooth or wired headset connected, all audio goes to the speaker, which is loud and is audible in noisy environments. If a headset is connected, the audio is routed to that.

5.18 Integration with existing time tracking application

A delivery company can integrate lone worker into their system. Assuming that they already have a Java application that is used to send messages to and from the server, they can integrate it with the lone worker system to do system checks, send location periodically and provide manual alarm.

The Java application can be used to enable or disable the lone worker man down mode. When the lone worker client is active, popup alerts can be sent to the user from the server. For example, if the Java application is used to send new tasks to the user, the lone worker alert can be used to indicate new tasks on the Java application. Seeing the alert, the user can open the Java application and check the task.

6 Functionality

6.1 Identification

A user id is used in all communication between the server and client. The server needs to configure the user id of the phone which is a number in the range of 1 to 4294967295 when sending the configuration. All messages from the client to the server have the IMEI number of the phone.

6.2 Lone Worker Modes

There are several modes of operation of the phone as below.

Icon	Mode name	Features available		
		GPS, Battery, Awake, E2E, Health checks, manual alarm	**Location tracking**	**Man down**
(no icon)	Disabled (0)	No	No	No
LE	Simple (1)	Yes	No	No
LT	Tracking (2)	Yes	Yes	No
LM	Man down (3)	Yes	Yes	Yes
LA	Alarm (4)	Yes	Yes	Depends

The phone has red, amber and green buttons to easily change the modes.

Long press of the red button will change the mode to *alarm* and raise manual alarm. This will work even when the keypad is locked. The numeric keys 2 and 3 are colored amber and green respectively. Long presses of these keys in the idle screen when the keypad is unlocked are mapped to Java applications.

The lone worker modes can be changed from the lone worker menu item. The menu item will be present only if it is enabled in the configuration. Long press of numeric key 6 will launch the menu is enabled. If a minimum mode is configured, then modes lower than that cannot be set.

Each feature has an individual enable/disable state apart from the over all Lone Worker mode. The Lone Worker modes other than Disabled can be generically called as Enabled modes. If the client is in Disabled mode, none of the checks are active, even if their individual state is enabled. If it is in one of the Enabled modes, the checks that are individually enabled are active. Similarly, the location tracking and man down also can be individually enabled or disabled.

If the user is manually changing the lone worker mode, he can set to disabled, simple, tracking or man down mode. The user cannot change any other parameters using the native application. All mode changes are reported to the server.

The mode can be changed by the server. When the mode is changed from the server, no confirmation is taken from the user.

Whenever the mode is changed from disabled to any other mode, an end-to-end network connectivity check is done and reported to the user.

When the phone is shutdown, an indication is sent to the server.

If only limited reports are opted for, then the end-to-end check is not done when the application is enabled. No report is generated when mode is changed or phone is shutdown.

6.3 Automatic Mode Change

Time can be set to change the mode automatically to and from the modes Tracking and Man down. A time value of 2400 indicates it is disabled.

If TrackingStartTime is set, at that time, if the current mode is less than Tracking, the mode is set to Tracking. The previous mode is remembered as the tracking stop mode. If TrackingStopTime is set, at that time, if the current mode is Tracking, the tracking stop mode is set as the current mode.

If MandownStartTime is set, at that time, if the current mode is less than Mandown, the mode is set to Mandown. The previous mode is remembered as the mandown stop mode. If MandownStopTime is set, at that time, if the current mode is Mandown, the mandown stop mode is set as the current mode.

If both TrackingStartTime and TrackingStopTime are set, when the phone is powered on, the mode is set depending on whether the current time is within the ON period or OFF period. Similarly the mode is set based on MandownStartTime and MandownStopTime also.

All mode changes are informed to the user by a local warning. The user is informed of the mode change.

6.4 Incident Reporting

When a situation that needs attention is detected, it can be reported in several ways. Here are the various reporting modes.

Mode	Name	Description
0	Local warning	This is for local warnings that are OK not to be noticed by the user. Raise a local alarm, which vanishes automatically if it is not

		acknowledged in 5 seconds (configurable). Long beep is played when the display comes up.
1	Local alarm	This is for local alarms that need to be noticed by the user. Raise a local alarm, which has to be acknowledged by the user. Long beep is played when the display comes up and a short beep is played again every 10 seconds till the user acknowledges.
2	Remote alarm	This is for alarms that are sent to the server if not canceled by the user. Raise a local alarm, which the user can disable or confirm within 10 seconds (configurable). If the user does not disable, send the alarm to the server. The mode is changed to alarm mode. Long beep is played when the display comes up and is played every 3 seconds till the alarm is sent to the server. There after, a short beep is played every 10 seconds.
3	Alarm with call	This is for alarms that are sent to the server if not canceled by the user, and also a GSM call to be made to the ERC. Raise a local alarm, which the user can disable or confirm within 10 seconds. If the user does not disable, send the alarm to the server and also make a GSM call to the first configured ERC number. If not reachable within 10 seconds (configurable), try the next number. Try the numbers a loop till one of them is reached or the user cancels the alarm. The mode is changed to alarm mode. Long beep is played when the display comes up and is played every 3 seconds till the alarm is sent to the server. There after, a short beep is played every 10 seconds.
4	Silent alarm	This is typically used when the user manually raises the alarm. Send the alarm to the server and also make a GSM call to the first configured ERC number. If not reachable, try the next number. This goes on in a loop till one of them is reached or the user cancels the alarm. All local sounds are disabled till the ERC is reached. The mode is changed to alarm mode. The call is put into speaker mode or normal mode depending on the configuration. CLIP is enabled.
5	Remote report	This is to send a report to the server without any confirmation from the user. The user will not be informed of this at all.
6	Remote info	This is to send information to the server without any confirmation from the user. The user will be informed of this.
7	No report	This is to do nothing about the incident.

The audio for the long beep and the short beep are configurable. All the time intervals are also configurable. Appropriate messages are displayed on the phone when the user is alerted.

6.5 GSM Check

If GSM check is enabled, GSM network coverage is checked periodically. If GSM coverage is not present, *local alarm* is raised.

6.6 GPS Check

If GPS check is enabled, GPS connectivity is checked periodically. If GPS connectivity is not present, *local warning* is raised.

6.7 Battery Check

If battery level check is enabled, battery level is checked periodically. If battery level is lower than the configured threshold, *local warning* is raised.

6.8 Awake Check

If awake check is enabled, *remote alarm* is raised periodically. Typically the user will cancel the alarm before the alarm is sent to the server.

6.9 E2E Check

If end-to-end check is enabled, round trip check is done to ensure the server is reachable from the client and the client is reachable from the server. If end-to-end check fails, *local alarm* is raised.

6.10 System Health Report

If sending health is enabled, the client sends a health report regularly as configured. This conforms to the DIN 0825-11 standard. If the health could be sent, the phone shows a *local warning*. If the health could not be sent, a *local alarm* is raised. The data sent to the server includes GPS location, Cell id, battery level, accelerometer values and connectivity to GPRS, GPS, etc.

6.11 Manual Alarm

If the user presses the alarm (red) button for more than the specified duration, *silent alarm* is raised. The alarm button can be used even when the keypad is locked.

6.12 Keypad Events

Long press of numeric buttons 2, 3, 4, 5, 6, 7, 8, 9 and 0 can be configured to send an event to the server. These can be used even when the keypad is locked.

6.13 Location Tracking

If location sending is enabled, location is sent periodically to the server.

6.14 Man Down

If tilt detection is enabled, if the phone is tilted beyond the configured threshold for more time than configured, *remote alarm with call* is raised. If idle detection is enabled, if the phone is idle for more than the time configured*, remote alarm with call* is raised. If impact detection is enabled, if the phone faces an impact more than the threshold configured*, remote alarm with call* is raised. If fall detection is enabled, if the phone falls freely*, remote alarm with call* is raised. A preconfigured message is displayed and a preconfigured audio is played before sending the alarm to the server and making a call to ERC. The user can cancel the alarm within the configured time duration before the alarm is sent to the server and call is made to ERC.

6.15 Transport Modes

Here are the modes to send messages to the server. Typically, the default transport is HTTP and the backup transport is SMS.

Mode	Name	Description
0	Try once	This is for messages that are just for information to the user. Try just once. If not possible to send now, just ignore the message. Don't check for response.
1	Try later	This is for messages that are OK to get lost under extreme situations. If there is no connectivity now, queue the message and try later. If the queue is getting full remove this message from the queue.
2	Send now or later	This is for important but non-urgent messages. If there is connectivity now, send the message and queue the message too. Remove the message from the queue if the message has been delivered successfully. If there is no response, try later. If there is no connectivity now, queue the message and try later. Never remove this message from the queue to make place for "Try later" messages. If the queue becomes full when there are no "Try later" messages, then the oldest message has to be removed.
3	Send now	This is for important and urgent messages. If there is connectivity now, send the message and queue the message too. Remove the message from the queue if the message has been delivered successfully. If there is no response, try the backup transport (typically SMS). If that also fails, then queue the message and try later. Every time, try the default transport (typically HTTP) first and then the

		backup transport (typically SMS). Never remove this message from the queue to make place for “Try later” messages. If the queue becomes full when there are no “Try later” messages, then the oldest message has to be removed.

The transport mode can be configured for different kind of messages.

6.16 Configuration

The XP3 Sentinel phone has three kinds of storage.

- The internal flash memory is not available for direct use by the user. It can be used only by native applications.
- The public flash memory is where files transferred by Bluetooth are stored. It can be accessed by the user using the *My Files* menu on the phone. This can be accessed by native and Java applications.
- The SD card memory is visible on Windows Explorer when the phone is connected to a PC using a standard USB cable. It can be accessed by the user using the *My Files* menu on the phone. This can be accessed by native and Java applications.

When the client starts up, it looks for configuration data in the following locations:

1. Internal flash memory
2. *LWConfig.txt* file in the *Others* folder in the SD card memory
3. *LWConfig.txt* file in the *Others* folder in the public flash memory

Data in the *Others* folder in the public flash memory overrides that in the SD card memory, which overrides that in the internal flash memory. Data in the public flash memory is considered only if it is newer than the previously loaded data from public flash memory. Data in the SD card memory is considered only if it is newer than the previously loaded data from SD card memory.

One or more parameters can be pushed from the server to the client by OTA SMS. The server can request the client to pull one or more or all parameters from the server. The server can request the client for one or more or all parameters.

No confirmation is asked from the user before updating the configuration based on OTA SMS.

The encryption key on the phone can be set for the first time over the air. It cannot be modified by over the air configuration. A native application started using a #5323# allows reading and updating the encryption key.

Parameter names are case insensitive. The names are in English only. Parameter values are in UTF-8.

6.17 GSM Call Interaction

GSM calls have lower priority than alarms on Lone Worker.

If there is an active GSM call, GPRS is not available. “Try once” messages are dropped. “Try later” and “Send now or later” messages are queued. “Send now” messages are sent over SMS. “Local warning” are deferred till the call ends. “Local alarm”, “Remote alarm”, “Remote alarm with call” and “Silent alarm” terminate the call, unless they are from or to one of the ERC numbers configured.

If there is an active GSM call with one of the ERC numbers, “Local alarm” and “Remote alarm” do everything as usual without ringing the alarm. No new GSM call is established for “Remote alarm with call” and “Silent alarm”.

When Lone Worker MMI for “Local warning” or any configuration is being shown, if there is an incoming GSM call, the Lone Worker MMI is closed. When Lone Worker MMI for “Local alarm”, “Remote alarm”, “Remote alarm with call” or “Silent alarm” is being shown and the Lone Worker is not in “alarm” mode, if there is an incoming GSM call, the GSM call is terminated.

When Lone Worker is in “alarm” mode, if there is an incoming GSM call from a number that is not one of the ERC numbers configured, the GSM call is terminated.

If there is an incoming GSM call from one of the ERC numbers configured, the GSM call is accepted automatically and the call is put into speaker mode.

When a call is placed to the ERC, there is a one second vibration if the call could be initiated. There is a long three second vibration is the call could not be placed because of no GSM coverage. There is a one second vibration when the call is connected to the ERC. Whenever the phone is in a call with ERC, the phone vibrates periodically to let the user know that he is connected to the ERC.

6.18 Java Interaction

The number keys 2 and 3 are colored amber and green respectively on XP3 Sentinel phones. These phones can be mapped to launch Java applications with MIDlet names Two and Three respectively. The Java applications can change the mode of the Lone Worker application by the Java class com.sonimtech.LoneWorker. This way, we can make the amber and green buttons behave as shortcuts to change the mode of the Lone Worker application based on some interaction with the user or a server. By default, the phone is shipped with simple implementations of MIDlets Two and Three, which do nothing other than changing the mode.

Java applications can also cause the Lone Worker application to dump its current configuration into a file and also update the current configuration from a file. By this, Java applications can change configuration parameters.

```
class LoneWorker
{
	public static final int DISABLED = 0;
	public static final int SIMPLE = 1;
	public static final int TRACKING = 2;
	public static final int MANDOWN = 3;
	public static final int ALARM = 4;

	public static final int SUCCESS = 200;
	public static final int BAD_ARGUMENT = 400;
	public static final int FORBIDDEN = 403;
	public static final int NOT_AVAILABLE = 404;
	public static final int UNKNOWN_ERROR = 500;

	public static int getMode ( );

	public static int setMode ( int mode );

	public static int getConfiguration ( String filename );

	public static int getDefinition ( String filename );

	public static int setConfiguration ( String filename );

};
```

A Java application is provided which uses this API to change configuration parameters and also helps to view and change the mode of the Lone Worker application. This Java application is highly customizable. The list of configurable parameters can be customized.

Sonim provides full technical support for partners to interface their Java applications to the Lone Worker API. Sonim has a portal http://www.sonimdevelopers.com to support partners developing Java applications for Sonim phones. Sonim has a customized J2ME SDK which can be used by partners to develop Java applications on the XP3 Sentinel.

The Java platform on XP3 Sentinel has advanced features like running a Java application in the background, automatically pushing a Java application into background when any other native application comes to foreground, auto start of a Java application on phone power up and defining a Java application as no-exit, in which case, the application cannot be terminated accidentally.

However, as the phone supports only one Java application to be running at anytime, if the amber and green keys are mapped to start Java applications, then there should not be any Java application running or suspended in the background. If there is any, then the phone will only ask the user if he wants to terminate or continue the application in the background. This also implies that the application that is mapped to the amber and green keys come up, interact with the user if necessary, do their job (like connect to network, sent SMS, change loneworker mode or/and configuration, etc) and terminate. They should not run in the background.

6.19 Sample Java Code

Here is a sample Java code that changes the mode of the Lone Worker client to Simple.

```
int result;
result = LoneWorker.setMode(LoneWorker.SIMPLE);

if(result != LoneWorker.SUCCESS)
{
…
}
```

There are sample applications in the folder *examples* in the FTP site.

7 Configuration Parameters

Here is the complete list of all configuration parameters.

The audio files in the SD card and Bluetooth transfer folder can be used for various audio indications. The SD card memory is accessible at the path “/rem1”. So the Sounds folder on SD card memory will be “/rem1/Sounds”. The Bluetooth transfer folder is accessible at the path “/app3/user”. So the Sounds folder on the Bluetooth transfer folder will be “/app3/user/Sounds”.

Parameter	Type	Description	Client configurable	Default	Example
UserId	Uint32	Identification of the user	No		20953
SystemName	String	Name of the lone worker system	No		Akela Hoon
UserName	String	Name of the user	No		Agent 007
EncryptionKey	String	Encryption key	No	disabled	
LogLevel	Uint16	Logging level. 0 implies no logs	Yes	0	
DataVersionPublic	Unit16	Version number of data taken from public flash area	No	0	
DataVersionSD	Uint16	Version number of data taken from SD card	No	0	
MenuEnabled	Boolean	Is Lone Worker menu enabled?	No	0	
HttpUrl	String	Http url to send all data to	Yes		http://myloneworker.com:8080/process.php
SmsHost	String	Comma separated list of MSISDN numbers to send SMS messages to. Maximum of 5 numbers are allowed.	Yes		+919844198441, +919844198442, +919844198443
SendVia	Values	0 => HTTP 1 => SMS 2 => either	Yes	2	

		3 => None			
RestartOnStuckMessage	Boolean	If the GPRS connection could not be recovered after coming back into coverage, even after several attempts, should the phone be restarted?	Yes	0	
QueueSize	Uint16	Total number of messages to queue if out of coverage. Maximum is 100.	No	1024	
Queue50Reporting	Values	One of the reporting modes to report 50% queue full	Yes	0	
Queue80Reporting	Values	One of the reporting modes to report 80% queue full	Yes	1	
Queue100Reporting	Values	One of the reporting modes to report 100% queue full	Yes	1	
RestartOnStuckMessage	Boolean	Automatically restart the phone if the GPRS re-connection wasn't successful even after 20 tries at 30 second interval	Yes	0	
LocalWarningDuration	Uint16	Number of seconds to ring local warning before taking the next action	Yes	5	
LocalAlarmDuration	Uint16	Number of seconds to ring local alarm before taking the next action	Yes	10	
LongBeepInterval	Uint16	Number of seconds duration and interval before repeating local alarm	Yes	3	

ShortBeepInterval	Uint16	Number of seconds interval before repeating local beep after activating the alarm	Yes	10	
LongBeepAudioMessage	String	Path to the audio file containing the audio message to be played repeatedly before activating the alarm	Yes		
ShortBeepAudioMessage	String	Path to the audio file containing the audio message to be played repeatedly after activating the alarm	Yes		
LimitedReports	Boolean	If to suppress all extra reports	Yes	0	
UseShortenedMessages	Boolean	If to use only important fields in messages	Yes	0	
IncludeTimeZone	Boolean	If to include offset parameter in messages	Yes	0	
ERCNumbers	String	Comma separated list of MSISDN numbers of ERC in the order of preference. Maximum of 5 numbers are allowed.	Yes		+919844198441, +919844198442, +919844198443
ERCTimeout	Unit16	Number of seconds to wait for each number of the ERC to respond to call	Yes	30	
UseSpeaker	Boolean	Should speaker be used for outbound calls to ERC?	Yes	1	
EnableCLIP	Boolean	Should CLIP be enabled before making call to	Yes	0	

		ERC?			
UseVibrator	Boolean	Should the phone vibrate when making calls to ERC to indicate the status?	Yes	1	
UseVibrationInCall	Boolean	Should the phone vibrate periodically when connected to ERC?	Yes	1	
VibrationInterval	Uint16	Number of seconds interval for vibration when in call with ERC	Yes	30	
LoneWorkerMode	Values	One of the lone worker modes	Yes	0	
LoneWorkerMinMode	Values	One of the configurable lone worker modes (LE, LT, LM)	Yes	1	
TrackingStartTime	Uint32	hhmm of day of start time. 2400 indicates disabled.	Yes	2400	1245
TrackingStopTime	Uint32	hhmm of day of stop time. 2400 indicates disabled.	Yes	2400	1345
MandownStartTime	Uint32	hhmm of day of start time. 2400 indicates disabled.	Yes	2400	1400
MandownStopTime	Uint32	hhmm of day of stop time. 2400 indicates disabled.	Yes	2400	1500
TrackingStopMode	Values	One of the lone worker modes that is to be set when stopping tracking.	Yes	1	
MandownStopMode	Values	One of the lone worker modes that is to be set when stopping mandown.	Yes	2	

HealthSend	Boolean	If to send health at regular intervals	Yes	0	
HealthSendInterval	Uint16	Interval of sending in seconds. Range 300:43200	Yes	3600	
HealthSendMode	Values	One of the transport modes. This mode is used to send all health and alarm messages.	Yes	1	
LocationSource	Values	0 => GPS 1 => CellId	Yes	0	
HealthSendTimeout	Uint16	Time in seconds to wait for health send response before alerting the user. Range 30:43200	Yes	60	
HealthSendSuccessReporting	Values	One of the reporting modes	Yes	7	
HealthSendFailureReporting	Values	One of the reporting modes	Yes	7	
GSMCheck	Boolean	If to check GSM coverage at regular intervals	Yes	0	
GSMCheckInterval	Uint16	Interval of checking in seconds. Range 300:43200	Yes	600	
GSMFailureReporting	Values	One of the reporting modes	Yes	1	
GSMFailureThreshold	Uint16	Lower threshold of percentage of signal strength	Yes	20	
GSMSignalThreshold	Unit16	Lower threshold of percentage of signal strength below which ERC calls should be failed.	Yes	10	
GPSCheck	Boolean	If to check GPS connectivity at	Yes	0	

		regular intervals			
GPSCheckInterval	Uint16	Interval of checking in seconds. Range 300:43200	Yes	600	
GPSFailureReporting	Values	One of the reporting modes	Yes	1	
GPSUpdateInterval	Uint16	Interval of updating GPS location in seconds. Range 2:43200	Yes	120	
CurrentLocationTimeout	Uint16	Number of seconds to wait for GPS fix when getting current location	Yes	30	
BatteryCheck	Boolean	If to check battery strength at regular intervals	Yes	0	
BatteryCheckInterval	Uint16	Interval of checking in seconds. Range 300:43200	Yes	600	
BatteryFailureReporting	Values	One of the reporting modes	Yes	1	
BatteryFailureThreshold	Unit16	Percentage of battery level below which alarm has to be raised	Yes	20	
AwakeCheck	Boolean	If to check the user is awake at regular intervals	Yes	0	
AwakeCheckInterval	Uint16	Interval of checking in seconds. Range 300:43200	Yes	3600	
AwakeFailureReporting	Values	One of the reporting modes	Yes	3	
AwakeCheckAudioMessage	String	Path to the audio file containing the audio message to be played repeatedly before activating the alarm	Yes		

AwakeCheckMessageInterval	Uint16	Interval and duration to play Audio message	Yes	10	
TechnicalCheckAudioMessage	String	Path to the audio file containing the audio message to be played repeatedly before activating the alarm	Yes		
TechnicalCheckMessageInterval	Uint16	Interval and duration to play Audio message	Yes	10	
E2ECheck	Boolean	If to check E2E connectivity at regular intervals	Yes	0	
E2ECheckInterval	Uint16	Interval of checking in seconds. Range 300:43200	Yes	600	
E2ETimeout	Uint16	Timeout value in seconds. Range 10:43200	Yes	60	
E2EFailureReporting	Values	One of the reporting modes	Yes	2	
PhoneStatus	Uint16	Bitmask of phone status as in Health report's src parameter. A changeconfig request for this is send by the phone for E2E check	No	0	
LocationSend	Boolean	If to send Location at regular intervals	Yes	1	
LocationSendInterval	Uint16	Interval of sending in seconds. Range 2:43200	Yes	600	
LocationSendIntervalAlarm	Uint16	Interval of sending in seconds in alarm mode. Range 2:43200	Yes	600	
LocationSendMode	Values	One of the transport	Yes	2	

		modes			
MandownCheckIdle	Boolean	If to check for idle at regular intervals	Yes	0	
MandownCheckTilt	Boolean	If to check for tilt at regular intervals	Yes	0	
MandownCheckImpact	Boolean	If to check for impact	Yes	1	
MandownCheckFall	Boolean	If to check for fall	Yes	1	
MandownIdleThreshold	Uint16	Acceleration threshold in percentage of 8G below which alarm should be raised	Yes	5	
MandownIdleTimeout	Uint16	Maxium duration in seconds for which the phone can be without movement. Range 10:43200	Yes	10	
MandownTiltThreshold	Uint16	Tilt threshold in degrees beyond which the phone is considered tilted	Yes	20	
MandownTiltTimeout	Unit16	Maximum duration in seconds for which the phone can be tilted. Range 10:43200	Yes	10	
MandownImpactThreshold	Uint16	Acceleration threshold in percentage of 8G beyond which alarm should be raised	Yes	80	
MandownImpactTimeout	Unit16	Maximum duration in milliseconds for which the phone should be beyond acceleration threshold. Range 10:43200	Yes	10	
MandownFallThreshold	Uint16	Acceleration threshold in percentage of 8G	Yes	3	

		below which alarm should be raised			
MandownFallTimeout	Unit16	Minimum duration in milliseconds for which the phone should fall. Range 10:43200	Yes	100	
MandownTextMessage	String	Text to be displayed before activating the alarm	Yes		
MandownAudioMessage	String	Path to the audio file containing the audio message to be played repeatedly before activating the alarm	Yes		
MandownMessageInterval	Uint16	Interval and duration to play Audio message	Yes	10	
MandownReporting	Values	One of the reporting modes	Yes	3	
ManualAlarm	Boolean	If to support manual alarm	Yes	1	
ManualAlarmDuration	Uint16	Number of seconds to press the alarm button to raise alarm. Range 2:10	Yes	5	
ManualAlarmReporting	Values	One of the reporting modes	Yes	4	
AmberKeyAlarm	Boolean	If to support amber key alarm	Yes	0	
AmberKeyAlarmReporting	Values	One of the reporting modes	Yes	5	
GreenKeyAlarm	Boolean	If to support green key alarm	Yes	0	
GreenKeyAlarmReporting	Values	One of the reporting modes	Yes	5	
NumberKeyAlarm	Boolean	If to support number key alarm for keys	Yes	0	

		4, 5, 6, 7, 8, 9, 0			
NumberKeyAlarmReporting	Values	One of the reporting modes	Yes	5	

When switching to Alarm mode, the phone stores the current mode into a parameter named PreviousMode. This can be read by server. Setting this parameter is not recommended.

8 Lone Worker Network Protocol

8.1 Introduction

8.1.1 Transport

The Lone Worker application on the phone interacts with the server using XML messages over HTTP POST or short text messages over SMS. The requests from the server are always over SMS. If HTTP port is not specified, port 80 is used.

8.1.2 Format

The order of the occurrence of tags in the XML messages is mandatory. The tags have no attributes. They have only CDATA values.

The text message is merely a concatenation of the CDATA parts in the XML message with # (hash) as the separator. All text messages end with #. Each CDATA part separated by # is referred to as a *word*. The complete text message should always be less than 160 bytes.

The encoding of both XML and text messages is UTF-8.

8.1.3 Encryption

All messages from and to the server are encrypted using a key that is configured both on the phone and on the server. This key on the phone cannot be modified over the air. If the key has the special value “disabled”, then encryption is turned off.

8.1.4 Messages

There are four kinds of messages.

Request – This is a request asking for some data.
Response – This is the response to the request.
Report – This is a report with some data.
Acknowledgement – This is the response to a report.

8.1.5 Message Direction

Unless otherwise specified, all request and acknowledgement are from the server to the mobile, and all response and report are from the mobile to the server.

8.1.6 Common Contents

Depending on the message, the enveloping tag in the XML is either <req>, <res>, <rep> or <ack>. This is the first word in the text message. This and the action are the only words that are case-insensitive.

The first tag inside the enveloping tag is always the <action> tag. This tells what the action or information is. The action value is case-insensitive. <action> tag is not used in <ack> messages. Other messages have <action> tag.

This is followed by the <ctag> tag. The sender of a <req> or <rep> generates this. The corresponding <res> or <ack> message has the same <ctag> value. This can be used to match the related messages. This is followed by the <uid> tag which gives the identity of the sender.

All messages that are from the client to the server have <imei> and <imsi> tags following the <uid> tag.

<status> tag is mandatory in <res> and <ack> messages. Note that the <status> tag is not related to the HTTP status, though the status number is similar.

<minutes> tag is mandatory in all messages from the server and from the client. This contains the number of minutes since the start of the UTC day. The server always sends its time here and the client keeps synchronizing with the server with every message from the server to the client. This tag is the last tag in all messages. This is used from HTTP messages. This is ignored by the client in SMS messages as the SMS can get delayed. But a dummy value has to be given for uniformity.

If the parameter IncludeTimeZone is set, an <offset> tag is inserted before the <minutes> tag with the offset between the server time (got from the minutes tag) and the phone's local clock time. It is presented in (-)hhmm format. This can be used as the approximate time zone of the phone.

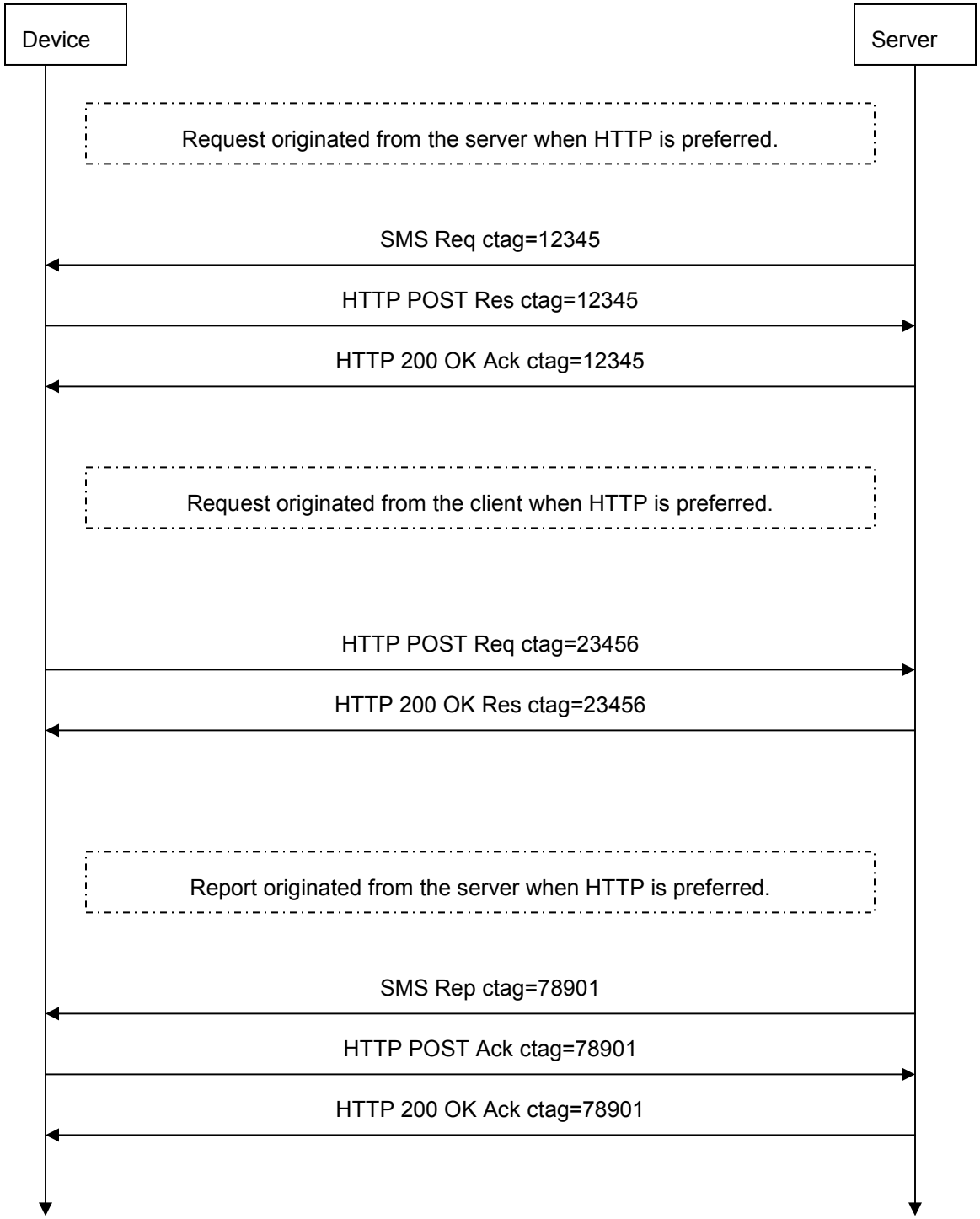
If the parameter UseShortenedMessages is set, some fields are removed or shortened to make sure that the SMS form of the messages are within 160 characters.

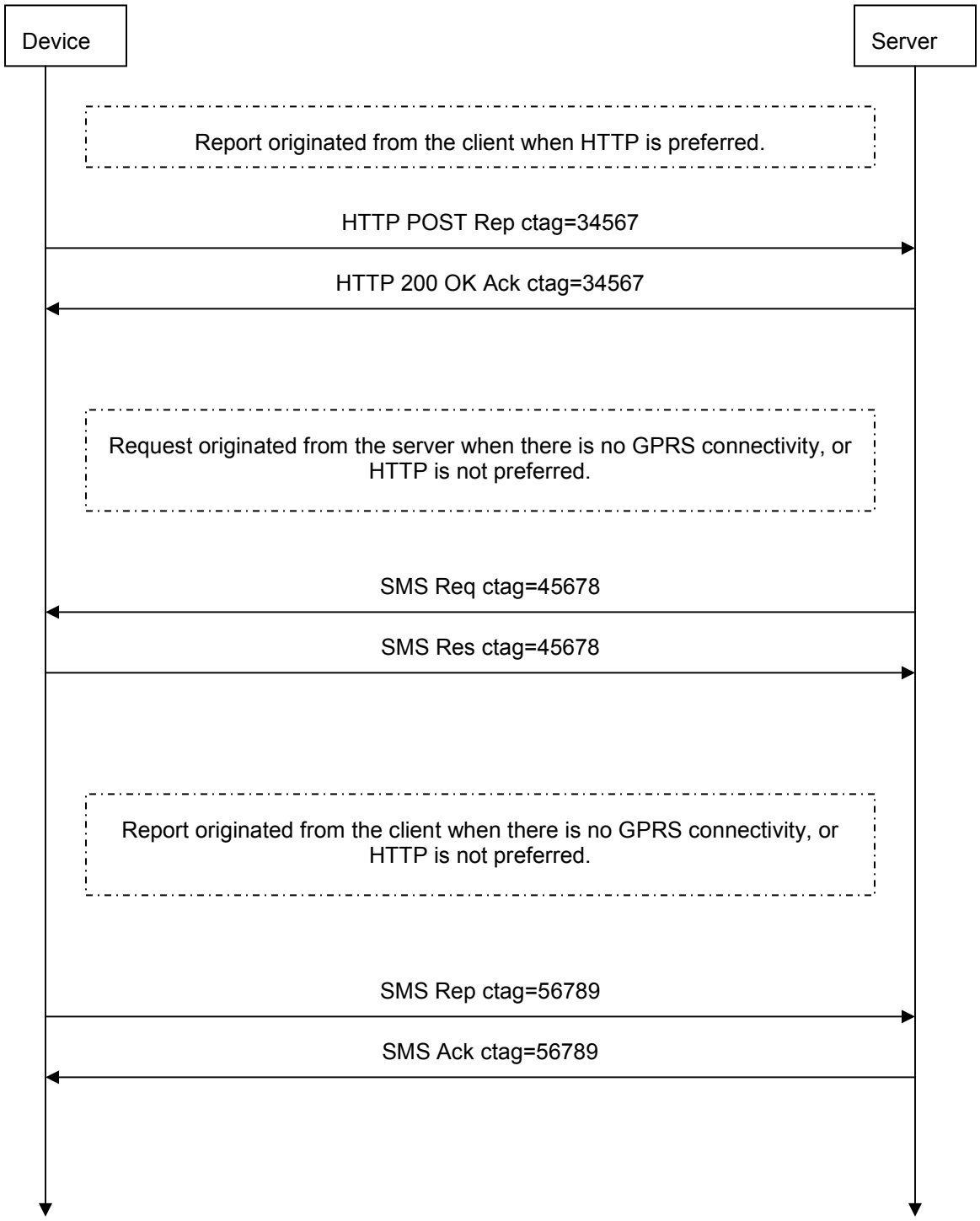
- IMSI is not sent in the messages.
- The first 5 digits of the IMEI is dropped.
- Accelerometer and Cell id values are not sent in health report and response.

The order of the tags is fixed. Please see the examples.

8.1.7 Message Sequences

The preferred mode of communication is HTTP. So if a server sends a request over SMS, the device sends the response over HTTP. Here are different scenarios.




Device
Server
Report originated from the client when HTTP is preferred.
HTTP POST Rep ctag=34567
HTTP 200 OK Ack ctag=34567
Request originated from the server when there is no GPRS connectivity, or HTTP is not preferred.
SMS Req ctag=45678
SMS Res ctag=45678
Report originated from the client when there is no GPRS connectivity, or HTTP is not preferred.
SMS Rep ctag=56789
SMS Ack ctag=56789

8.1.8 Examples

Here is an example of a request message with a missing optional parameter:

```
<? xml version="1.0" encoding="utf-8" ?>
<req>
<action>location</action>
<ctag>20090</ctag>
<uid>1</uid>
<mode>current</mode>
<timeout></timeout>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
REQ#location#20090#1#current##345#
```

Here is an example of a response message which has a few parameters:

```
<? xml version="1.0" encoding="utf-8" ?>
<res>
<action>location</action>
<ctag>20090</ctag>
<uid>78965</uid>
<imei>123456789012345</imei>
<imsi>123456789012345</imsi>
<status>200</status>
<lat>17.789653</lat>
<lon>79.459786</lon>
<hdop>1.2</hdop>
<alt>987.5</alt>
<spd>5.897</spd>
<dir>275.6</dir>
<time>20090726185403</time>
<minutes>345</minutes>
</res>
```

Here is the text form of the same message:

```
RES#location#20090#78965#123456789012345#123456789012345#200#17.789653#79.4597
86#1.2#987.5#5.897#275.6#20090726185403#345#
```

Here is an example of a report message:

```
<?xml version="1.0" encoding="utf-8"?>
<rep>
<action>health</action>
<ctag>4949</ctag>
<uid>92</uid>
<imei>356789431226188</imei>
<imsi>234456789796861</imsi>
<status>200</status>
<emg>0</emg><src>0</src><mode>2</mode>
<gsm>Y</gsm><gps>Y</gps><gprs>Y</gprs><bat>66</bat>
<x>6</x><y>-5</y><z>-72</z>
<lat>51.358176</lat><lon>-0.779528</lon><hdop>2.6</hdop>
<alt>9.72</alt><spd>0.68</spd><dir>23.92</dir><time>20100527111145</time>
<mcc>234</mcc><mnc>10</mnc><lac>14345</lac><cellid>21410</cellid>
<minutes>675</minutes>
</rep>
```

Here is the text form of the same message:

```
REP#health#4949#92#356789431226188#234456789796861#200#0#0#2#Y#Y#Y#66#6#-5#-72
#51.358176#-0.779528#2.6#9.72#0.68#23.92#20100527111145#234#10#14345#21410
#675#
```

Here is an example of an ack response to the above message:

```
<?xml version="1.0" encoding="UTF-8"?>
<ack>
<ctag>4949</ctag>
<uid>92</uid>
<status>200</status>
<minutes>675</minutes>
</ack>
```

Here is the text form of the same message:

```
ACK#4949#92#200#675#
```

8.1.9 Encryption Algorithm

Assumptions:

The key is always smaller than the payload.
The payload is a NULL terminated UTF-8 encoded C string.
The payload does not have these five characters: ^ | $ % *
The key has only alpha-numeric characters.

Encryption:

Let is say the key is 12345 and the payload is abcdefghijklmnopq.
The buffer is broken into chunks of the same size as the key.
abcde fghij klmno pq

Now the each chunk is XORed with the key.
ABCDE FGHIJ KLMNO PQ
If the values ‘, “, 00, FE and FF are found in the encrypted payload, those characters are replaced by ^, |, $, % and * respectively and encrypted.

Now consecutive chunks are swapped.
FGHIJ ABCDE PQ KLMNO
Each chunk is now reversed.
JIHGF EDCBA QP ONMLK
This is the encrypted payload.

Decryption:

Let us say the key is 12345 and the payload is ABCDEFGHIJKLMNOPQ.
The buffer is broken into chunks of the same size as the key, except the last but one chunk.
ABCDE FGHIJ KL MNOPQ
Each chunk is reversed.
EDCBA JIHGF LK QPONM
Now the consecutive chunks are swapped.
JIHGF EDCBA QPONM LK

Now each chunk is XORed with the key.
If characters ^, |, $, % and * are found in the decrypted payload, those encrypted charaters are replaced by ‘, “, 00, FE and FF respectively and decrypted.
jihgf edcba qponm lk
This is the decrypted payload.

Note:

The payload and the encrypted payload have the same number of characters.
The encrypted payload has only valid UTF-8 characters, but does not represent a valid Unicode string. So it should not be converted into Unicode or any other format or validated.

Please contact Sonim for the C code of encryption and decryption functions and the PHP extension wrapping them into a library.

8.1.10 PROV 1.0 SMS Message

The message from lone worker server to phone can come in OMA PROV 1.0 format. This message will be handled and used by the Lone Worker application. Here is the example of the message used.

```
<wap-provisioningdoc>
<characteristic type="APPLICATION">
  <parm name="APPID" value="LW_SMS" />
  <parm name="LW_APP_DATA" value="REQ#location#2345#1#current##345#" />
  </characteristic>
</wap-provisioningdoc>
```

This has to be compressed as per wbxml standard and sent to port **2948**. Note that the value of the parameter will be in encrypted form.

For convenience, here is a PHP code snippet with the readymade wbxml compression.

```
$udh = "\006\005\004\013\204\043\360";
$str = "REQ#location#2345#1#current##345#";
$encoded_str =
"\124\006\001\266\003\013j\000E\306\000\001U\001\2076\000\000\006\003LW_SMS\00
0\001\207\005\003LW_APP_DATA\000\006\003".$str."\000\001\001\001";
```

Currently, long SMS of up to about 1000 bytes are supported in this procedure.

However, SMS messages from the phone to the server will be in plain text form as below.

8.1.11 Plain Text SMS Message

Plain text SMS messages coming to the default SMS port are also processed by the Lone Worker application if they start with SONIMLW1 or SONIMLW2. The first format cannot be used with encryption enabled.

If the message starts with SONIMLW1, then the text message follows. Eg.
SONIMLW1#REQ#location#2345#1#current##345#

If the message starts with SONIMLW2, then the text message follows in base64 encoded form as per RFC 3548. Eg.
SONIMLW2#UkVRI2xvY2F0aW9uIzIzNDUjMSNjdXJyZW50IyMzNDUj==

If encryption is disabled, all SMS from phone to server will be of SONIMLW1 format. If encryption is enabled, all SMS from phone to server will be of SONIMLW2 format.

Currently, long SMS more than 160 characters are not supported in this procedure.

8.1.12 CP 1.1 SMS Message to set APN

The APN on the phone can be set by OTA SMS in CP 1.1 format. Here is the example of the message used.

```
<wap-provisioningdoc>
        <characteristic type="APPLICATION">
                <parm name="APPID" value="j1" />
                <parm name="TO-NAPID" value="APN_ENTRY" />
        </characteristic>
        <characteristic type="NAPDEF">
                <parm name="NAPID" value="APN_ENTRY"/>
                <parm name="NAME" value="My APN"/>
                <parm name="NAP-ADDRESS" value="HOSTNAME.COM"/>
                <parm name="BEARER" value="GSM-GPRS"/>
                <characteristic type="NAPAUTHINFO">
                        <parm name="AUTHTYPE" value="PAP"/>
                        <parm name="AUTHNAME" value="USERNAME"/>
                        <parm name="AUTHSECRET" value="PASSWORD"/>
                </characteristic>
        </characteristic>
</wap-provisioningdoc>
```

The values in **bold** have to be replaced by the correct values. This has to be compressed as per wbxml standard and sent to port **2948**.

8.1.13 Plain SMS to set APN

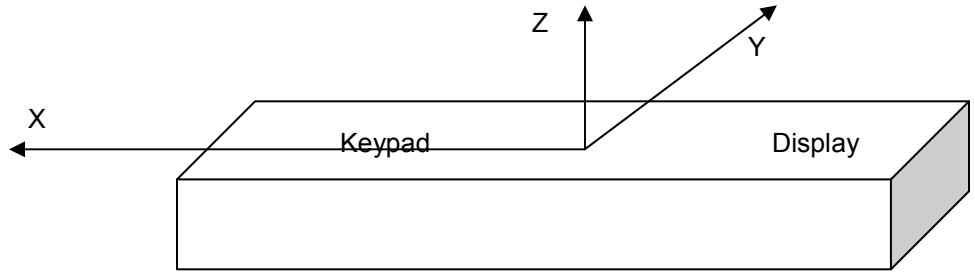
The APN on the phone can be set by OTA SMS in plain text format. Here is the example of the message used.

```
XP11|Network Profile 5|apn_name|user_name|user_pwd|5
```

The values in **bold** have to be replaced by the correct values.

8.1.14 Phone Orientation

Here are the directions of the phone for accelerometer:



8.2 Location

8.2.1 Location Request

Tag	Description	Occurrence	Example
Mode	last – Send last known value current – Send the current value	Mandatory	last
Timeout	Timeout value in seconds for mode “current”.	Optional	10

Fetch the last known or current location. If fetching current location, if timeout has occurred, send back 408 as status with the last known location values. A mobile unit can override the timeout value if it is outside of its supported range. If there is no GPS coverage currently also, send back the last known location values.

Here is an example. Note that as this is a message from server to phone, the XML form will never be used. It is given here for clarity.

```
<? xml version="1.0" encoding="utf-8" ?>
<req>
<action>location</action>
<ctag>20090</ctag>
<uid>1</uid>
<mode>current</mode>
<timeout>20</timeout>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
REQ#location#20090#1#current#20#345#
```

8.2.2 Location Response

Tag	Description	Occurrence	Example
Status	200 – Success 400 – Syntax error 403 – Another request pending 408 – Timeout 404 – No GPS coverage 500 – GPS not enabled 503 – No GPS fix yet	Mandatory	200
Lat	Latitude in [-]dd.dddddd format	Mandatory if status is not 400	17.789653

Lon	Longitude in [-]ddd.dddddd format	Mandatory if status is not 400	79.459786
Hdop	Horizontal dilution of precision in d.d format	Mandatory if status is not 400	1.2
Alt	Altitude in meters in d.d format	Mandatory if status is not 400	980.5
Spd	Speed in m/s	Mandatory if status is not 400	5.897
Dir	Direction of movement in ddd.dd format	Mandatory if status is not 400	275.6
Time	UTC timestamp in yyyymmddhhmmss format from satellite	Mandatory if status is not 400	20090726185403

Note that this format is different from the NMEA format. If the response is not 200 (or 400), the last known value is sent.

Here is an example:

```
<? xml version="1.0" encoding="utf-8" ?>
<res>
<action>location</action>
<ctag>20090</ctag>
<uid>78965</uid>
<imei>123456789012345</imei>
<imsi>123456789012345</imsi>
<status>200</status>
<lat>17.789653</lat>
<lon>79.459786</lon>
<hdop>1.2</hdop>
<alt>987.5</alt>
<spd>5.897</spd>
<dir>275.6</dir>
<time>20090726185403</time>
<minutes>345</minutes>
</res>
```

Here is the text form of the same message:

```
RES#location#20090#78965#123456789012345#123456789012345#200#17.789653#79.4597
86#1.2#987.5#5.897#275.6#20090726185403#345#
```

8.2.3 Location Report

Tag	Description	Occurrence	Example

Status	200 – GPS coverage present 404 – No GPS coverage 500 – GPS not enabled 503 – No GPS fix yet	Mandatory	200
Lat	Latitude in [-]dd.dddddd format	Mandatory	17.789653
Lon	Longitude in [-]ddd.dddddd format	Mandatory	79.459786
Hdop	Horizontal dilution of precision in d.d format	Mandatory	1.2
Alt	Altitude in meters in d.d format	Mandatory	980.5
Spd	Speed in m/s	Mandatory	5.897
Dir	Direction of movement in ddd.dd format	Mandatory	275.6
Time	UTC timestamp in yyyymmddhhmmss format from satellite	Mandatory	20090726185403

Note that this format is different from the NMEA format. If there is no GPS coverage, the values may not be recent.

Here is an example:

```
<? xml version="1.0" encoding="utf-8" ?>
<rep>
<action>location</action>
<ctag>20090</ctag>
<uid>78965</uid>
<imei>123456789012345</imei>
<imsi>123456789012345</imsi>
<status>200</status>
<lat>17.789653</lat>
<lon>79.459786</lon>
<hdop>1.2</hdop>
<alt>987.5</alt>
<spd>5.897</spd>
<dir>275.6</dir>
<time>20090726185403</time>
<minutes>345</minutes>
</rep>
```

Here is the text form of the same message:

```
REP#location#20090#78965#123456789012345#123456789012345#200#17.789653#79.4597
86#1.2#987.5#5.897#275.6#20090726185403#345#
```

8.2.4 Location Acknowledgement

Tag	Description	Occurrence	Example
Status	200 – Success 400 – Syntax error	Mandatory	200

Here is an example of an ack response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ack>
<ctag>4949</ctag>
<uid>92</uid>
<status>200</status>
<minutes>675</minutes>
</ack>
```

Here is the text form of the same message:

```
ACK#4949#92#200#675#
```

8.2.5 CellId Request

No parameters. Send the current Cell Id details.

Here is an example. Note that as this is a message from server to phone, the XML form will never be used. It is given here for clarity.

```
<? xml version="1.0" encoding="utf-8" ?>
<req>
<action>cellid</action>
<ctag>20090</ctag>
<uid>1</uid>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
REQ#cellid#20090#1#345#
```

8.2.6 CellId Response

Tag	Description	Occurrence	Example
Status	200 – Success 404 – Not connected	Mandatory	200
Mcc	MCC of the current cell	Mandatory	234
Mnc	MNC of the current cell	Mandatory	10

Lac	LAC of the current cell	Mandatory	2072
CellId	Cell Id of the current cell	Mandatory	12312

If there is no GSM coverage currently, send 404 as the status. The last known values are returned in that case.

Here is an example:

```
<? xml version="1.0" encoding="utf-8" ?>
<res>
<action>cellid</action>
<ctag>20090</ctag>
<uid>78965</uid>
<imei>123456789012345</imei>
<imsi>123456789012345</imsi>
<status>200</status>
<mcc>234</mcc>
<mnc>10</mnc>
<lac>14345</lac>
<cellid>21410</cellid>
<minutes>345</minutes>
</res>
```

Here is the text form of the same message:

```
RES#cellid#20090#78965#123456789012345#123456789012345#200#234#10#14345#21410#
345#
```

8.2.7 CellId Report

Tag	Description	Occurrence	Example
Status	200 – Connected 404 – Not connected	Mandatory	200
Mcc	MCC of the current cell	Mandatory	234
Mnc	MNC of the current cell	Mandatory	10
Lac	LAC of the current cell	Mandatory	2072
CellId	Cell Id of the current cell	Mandatory	12312

If there is no GSM coverage currently, send 404 as the status. The last known values are returned in that case. It may not be recent.

Here is an example:

```
<? xml version="1.0" encoding="utf-8" ?>
<rep>
<action>cellid</action>
<ctag>20090</ctag>
<uid>78965</uid>
<imei>123456789012345</imei>
<imsi>123456789012345</imsi>
<status>200</status>
<mcc>234</mcc>
<mnc>10</mnc>
<lac>14345</lac>
<cellid>21410</cellid>
<minutes>345</minutes>
</rep>
```

Here is the text form of the same message:

```
REP#cellid#20090#78965#123456789012345#123456789012345#200#234#10#14345#21410#
345#
```

8.2.8 CellId Acknowledgement

Tag	Description	Occurrence	Example
Status	200 – Success 400 – Syntax error	Mandatory	200

Here is an example of an ack response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ack>
<ctag>4949</ctag>
<uid>92</uid>
<status>200</status>
<minutes>675</minutes>
</ack>
```

Here is the text form of the same message:

```
ACK#4949#92#200#675#
```

8.3 Health and Emergency

8.3.1 Health Request

No parameters. Send the current health details.

Here is an example. Note that as this is a message from server to phone, the XML form will never be used. It is given here for clarity.

```
<? xml version="1.0" encoding="utf-8" ?>
<req>
<action>health</action>
<ctag>20090</ctag>
<uid>1</uid>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
REQ#health#20090#1#345#
```

8.3.2 Health Response

Tag	Description	Occurrence	Example
Status	200 – Success	Mandatory	200
Emg	Emergency level. 0 => all OK 1 => Attention needed but not urgent 2 => Attention needed immediately 3 => Full alert situation	Mandatory	0
Src	Emergency situation bit mask 0 => Red key pressed 1 => Phone problem 2 => Network problem 3 => GPS problem 4 => Phone idle 5 => High impact 6 => Free fall 7 => Phone tilt 8 => End to end disconnect 9 => Not awake 10 => Server request 11 => Shutdown 12 => Mode change 13 => Amber key (number key 2) pressed 14 => Green key (number key 3) pressed 15 => Number key 0 pressed 16 => Number key 1 pressed (not used) 17 => Number key 4 pressed 18 => Number key 5 pressed 19 => Number key 6 pressed 20 => Number key 7 pressed 21 => Number key 8 pressed 22 => Number key 9 pressed	Mandatory if Emg is not 0	b000000001

Mode	Lone worker mode	Mandatory	0
Gsm	Y – GSM connected N – GSM not connected	Mandatory	Y
Gps	Y – GPS fix present N – GPS fix not present D – GPS disabled	Mandatory	N
Gprs	Y – GPRS connected N – GPRS not connected	Mandatory	Y
Bat	Battery status in percentage of remaining power	Mandatory	75
X	Acceleration on X direction in m/s^2 in [-]dd.dd format	Mandatory	9.23
Y	Acceleration on Y direction in m/s^2 in [-]dd.dd format	Mandatory	9.23
Z	Acceleration on Z direction in m/s^2 in [-]dd.dd format	Mandatory	9.23
Lat	Latitude in [-]dd.dddddd format	Mandatory	17.789653
Lon	Longitude in [-]ddd.dddddd format	Mandatory	79.459786
Hdop	Horizontal dilution of precision in d.d format	Mandatory	1.2
Alt	Altitude in meters in d.d format	Mandatory	980.5
Spd	Speed in m/s	Mandatory	5.897
Dir	Direction of movement in ddd.dd format	Mandatory	275.6
Time	UTC timestamp in yyyymmddhhmmss format from satellite	Mandatory	20090726185403
Mcc	MCC of the current cell	Mandatory	234
Mnc	MNC of the current cell	Mandatory	10
Lac	LAC of the current cell	Mandatory	2072
CellId	Cell Id of the current cell	Mandatory	12312

Last known GPS and cell information are sent if they are not available. Accelerometer and cellid tags are dropped if UseShortenedMessages is set.

Here is an example of a response message:

```
<?xml version="1.0" encoding="utf-8"?>
<res>
<action>health</action>
<ctag>4949</ctag>
<uid>92</uid>
<imei>356789431226188</imei>
<imsi>234456789796861</imsi>
<status>200</status>
<emg>0</emg><src>0</src><mode>2</mode>
<gsm>Y</gsm><gps>Y</gps><gprs>Y</gprs><bat>66</bat>
<x>6</x><y>-5</y><z>-72</z>
<lat>51.358176</lat><lon>-0.779528</lon><hdop>2.6</hdop>
<alt>9.72</alt><spd>0.68</spd><dir>23.92</dir><time>20100527111145</time>
<mcc>234</mcc><mnc>10</mnc><lac>14345</lac><cellid>21410</cellid>
<minutes>675</minutes>
</res>
```

Here is the text form of the same message:

```
RES#health#4949#92#356789431226188#234456789796861#200#0#0#2#Y#Y#Y#66#6#-5#-72
#51.358176#-0.779528#2.6#9.72#0.68#23.92#20100527111145#234#10#14345#21410
#675#
```

8.3.3 Health Report and Emergency Alarm

Tag	Description	Occurrence	Example
Status	200 – Success	Mandatory	200
Emg	Emergency level. 0 => all OK 1 => Attention needed but not urgent 2 => Attention needed immediately 3 => Full alert situation	Mandatory	0
Src	Emergency situation bit mask 0 => Red key pressed 1 => Phone problem 2 => Network problem 3 => GPS problem 4 => Phone idle 5 => High impact 6 => Free fall 7 => Phone tilt 8 => End to end disconnect 9 => Not awake 10 => Server request 11 => Shutdown 12 => Mode change 13 => Amber key (number key 2) pressed 14 => Green key (number key 3) pressed	Mandatory if Emg is not 0	b000000001

	15 => Number key 0 pressed 16 => Number key 1 pressed (not used) 17 => Number key 4 pressed 18 => Number key 5 pressed 19 => Number key 6 pressed 20 => Number key 7 pressed 21 => Number key 8 pressed 22 => Number key 9 pressed		
Mode	Lone worker mode	Mandatory	0
Gsm	Y – GSM connected N – GSM not connected	Mandatory	Y
Gps	Y – GPS fix present N – GPS fix not present D – GPS disabled	Mandatory	N
Gprs	Y – GPRS connected N – GPRS not connected	Mandatory	Y
Bat	Battery status in percentage of remaining power	Mandatory	75
X	Acceleration on X direction in m/s^2 in [-]dd.dd format	Mandatory	9.23
Y	Acceleration on Y direction in m/s^2 in [-]dd.dd format	Mandatory	9.23
Z	Acceleration on Z direction in m/s^2 in [-]dd.dd format	Mandatory	9.23
Lat	Latitude in [-]dd.dddddd format	Mandatory	17.789653
Lon	Longitude in [-]ddd.dddddd format	Mandatory	79.459786
Hdop	Horizontal dilution of precision in d.d format	Mandatory	1.2
Alt	Altitude in meters in d.d format	Mandatory	980.5
Spd	Speed in m/s	Mandatory	5.897
Dir	Direction of movement in ddd.dd format	Mandatory	275.6
Time	UTC timestamp in yyyymmddhhmmss format from satellite	Mandatory	20090726185403
Mcc	MCC of the current cell	Mandatory	234
Mnc	MNC of the current cell	Mandatory	10

Lac	LAC of the current cell	Mandatory	2072
CellId	Cell Id of the current cell	Mandatory	12312

Last known GPS and cell information are sent if they are not available. Accelerometer and cellid tags are dropped if UseShortenedMessages is set.

Here is an example of a report message:

```
<?xml version="1.0" encoding="utf-8"?>
<rep>
<action>health</action>
<ctag>4949</ctag>
<uid>92</uid>
<imei>356789431226188</imei>
<imsi>234456789796861</imsi>
<status>200</status>
<emg>0</emg><src>0</src><mode>2</mode>
<gsm>Y</gsm><gps>Y</gps><gprs>Y</gprs><bat>66</bat>
<x>6</x><y>-5</y><z>-72</z>
<lat>51.358176</lat><lon>-0.779528</lon><hdop>2.6</hdop>
<alt>9.72</alt><spd>0.68</spd><dir>23.92</dir><time>20100527111145</time>
<mcc>234</mcc><mnc>10</mnc><lac>14345</lac><cellid>21410</cellid>
<minutes>675</minutes>
</rep>
```

Here is the text form of the same message:

```
REP#health#4949#92#356789431226188#234456789796861#200#0#0#2#Y#Y#Y#66#6#-5#-72
#51.358176#-0.779528#2.6#9.72#0.68#23.92#20100527111145#234#10#14345#21410
#675#
```

8.3.4 Health Acknowledgement

Tag	Description	Occurrence	Example
Status	200 – Success 400 – Syntax error	Mandatory	200

Here is an example of an ack response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ack>
<ctag>4949</ctag>
<uid>92</uid>
<status>200</status>
<minutes>675</minutes>
</ack>
```

Here is the text form of the same message:

```
ACK#4949#92#200#675#
```

8.4 Configuration

8.4.1 GetConfiguration Request

Tag	Description	Occurrence	Example
Params	Comma separated list of parameters or “all” to get all the parameters	Mandatory	all

This can be sent by the mobile to the server or the server to the mobile.

IMEI and IMSI will be present if it is from phone to server. They will not be present if it is from server to phone.

Here is an example:

```
<? xml version="1.0" encoding="utf-8" ?>
<req>
<action>getconfig</action>
<ctag>20090</ctag>
<uid>1</uid>
<imei>356789431226188</imei>
<imsi>234456789796861</imsi>
<params>UserId,LoneWorkerMode</params>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
REQ#getconfig#20090#1#356789431226188#234456789796861#UserId,LoneWorkerMode#34
5#
```

8.4.2 GetConfiguration Response

Tag	Description	Occurrence	Example
Status	200 – Success 400 – Syntax error	Mandatory	200
Name	Parameter name	Mandatory	LocationSendInterval
Value	Parameter value	Mandatory	30

There are as many <name> and <value> tags as there are parameters in the response. If a parameter is not set, empty value is returned for it.

This can be sent by the mobile to the server or the server to the mobile.

IMEI and IMSI will be present if it is from phone to server. They will not be present if it is from server to phone.

Here is an example:

```
<? xml version="1.0" encoding="utf-8" ?>
<res>
<action>getconfig</action>
<ctag>20090</ctag>
<uid>1</uid>
<status>200</status>
<name>UserId</name>
<value>24</value>
<name>LoneWorkerMode</name>
<value>2</value>
<minutes>345</minutes>
</res>
```

Here is the text form of the same message:

```
RES#getconfig#20090#1#200#UserId#24#LoneWorkerMode#2#345#
```

8.4.3 ChangeConfiguration Request

Tag	Description	Occurrence	Example
Name	Parameter name	Mandatory	LocationSendInterval
Value	Parameter value	Mandatory	30

There are as many <name> and <value> tags as there are parameters in the request. It is not possible to delete a parameter.

This is sent from the server to the mobile.

A request to set a special parameter “GetConfiguration” with value as a comma separated list of parameters, makes the mobile to send a “GetConfiguration Request” to the server with that parameter in the “Params” tag.

Here is an example.

```
<? xml version="1.0" encoding="utf-8" ?>
<req>
<action>changeconfig</action>
<ctag>20090</ctag>
<uid>1</uid>
<name>UserId</name>
<value>24</value>
<name>LoneWorkerMode</name>
<value>2</value>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
REQ#changeconfig#20090#1#UserId#24#LoneWorkerMode#2#345#
```

8.4.4 ChangeConfiguration Response

Tag	Description	Occurrence	Example
Status	200 – Success 400 – Syntax error	Mandatory	200

This is sent from the mobile to the server.

Here is an example:

```
<? xml version="1.0" encoding="utf-8" ?>
<res>
<action>changeconfig</action>
<ctag>20090</ctag>
<uid>1</uid>
<imei>356789431226188</imei>
<imsi>234456789796861</imsi>
<status>200</status>
<minutes>345</minutes>
</res>
```

Here is the text form of the same message:

```
RES#changeconfig#20090#1#356789431226188#234456789796861#200#345#
```

8.5 Alert

8.5.1 Alert Request

Tag	Description	Occurrence	Example
Title	Title of the alert	Mandatory	Caution
Body	Body of the alert	Mandatory	Man eater on the prowl.
Sk1	Soft key option 1	One of them is mandatory	Cancel
Sk2	Soft key option 2		OK
Timeout	Seconds to timeout and close the alert dialog. 0 => no timeout	Mandatory	0
Alarm	Should audible alert be raised?	Mandatory	Y

Duration	Audible alert duration in milliseconds. 0 => Default alarm duration.	Mandatory	10000

A MMI screen is shown to the user. Alarm is raised if specified. User response is sent back.

Here is an example. Note that as this is a message from server to phone, the XML form will never be used. It is given here for clarity.

```
<? xml version="1.0" encoding="utf-8" ?>
<req>
<action>alert</action>
<ctag>20090</ctag>
<uid>1</uid>
<title>Attention</title>
<body>Do you need help?</body>
<sk1>No</sk1>
<sk2>Yes</sk2>
<timeout>20</timeout>
<alarm>Y</alarm>
<duration>10000</duration>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
REQ#alert#20090#1#Attention#Do you need help?#No#Yes#20#Y#10000#345#
```

8.5.2 Alert Response

Tag	Description	Occurrence	Example
Status	200 – Success 400 – Syntax error	Mandatory	200
Event	Event to stop the alert 0 => User pressed Sk1 1 => User pressed Sk2 2 => Timeout 3 => Closed by another application 4 => Cannot display now 5 => Unexpected error	Mandatory	2

Here is an example.

```
<? xml version="1.0" encoding="utf-8" ?>
<res>
<action>alert</action>
<ctag>20090</ctag>
<uid>1</uid>
<imei>356789431226188</imei>
<imsi>234456789796861</imsi>
<status>200</status>
<event>1</event>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
RES#alert#20090#1#356789431226188#234456789796861#200#1#10000#345#
```

9 Best Practices

Here are some notes to get the best out of the Sonim Sentinel phone.

9.1 For End Users

These best practices are to be conveyed to the end users by the integrator.

1. For more stability, the keytone should be set to off. This can be done at Menu > Settings > Sounds > Key tone > Off

2. For seamless operation when moving across cell boundaries, the GPRS should be set to “always attached”. Menu > Settings > Connectivity > Network settings > Attach GPRS > Always attached

3. Please watch out for the GPS icon. If it is amber, the accuracy is poor. If it is green, the accuracy is OK. If it is red, it has not connected to the satellites. It may take more time to connect to the satellites if you are in non-ideal conditions like moving in a car or inside buildings. Always make sure that phone has maximum satellite view where ever possible, for .e.g.
    - Make sure the GPS icon is green before starting to drive
    - While in moving a car, always keep the phone on the dash board
    - Carry the phone always on the belt holster instead of keeping it in the pocket
    - Phone being in the vertical position gives better GPS performance

4. If you receive an incoming call from a non-ERC number after you’ve pressed the red-button, do not accept the call. In this case, after the LW icon on phone’s screen has changed to alarm icon, the ERC call will start automatically after about 30 seconds.

5. If you see “SS Error” or “Not allowed” when the phone is making call to ERC, don’t worry. The ERC call is attempted after 30 seconds

6. Always remove the phone from wall charger or USB connector when switching on or off the phone.

7. Using WAP browser has shown some interactions issues with the LW functionality. Recommendation is to use “Opera Mini” to browse internet. Avoid browsing internet for longer durations.

8. Java applications that use the Camera and Audio recorder are incompatible with reliable lone worker operation. Please do not use such applications.

9. When the phone is powered on, it will take about 10 to 15 seconds for the loneworker application to startup and be fully operational. Please wait for 15 seconds before placing any outgoing GSM calls. During this time, you can accept incoming GSM calls. In case, if you hear continuous beeps (E2E check failure on the screen) then power down the phone and restart the phone.

10. The red button is active in the idle screen even when keypad is locked. It is not active in other screens. If any other screen is displayed, then press the “END” key twice and come to the phone’s home screen before pressing the Red-Button.

9.2 For Integrators

These are some hints for the integrators.

1. Intervals for checks:
   Each cycle of local warning that display user prompts and the corresponding time out – in case user doesn't respond to the prompts – takes about 60 seconds. So it is recommended to use intervals more than 90 seconds for checks like GPS check, Awake check, Battery check, GSM check, etc.

2. If the phone is not able to connect to GPRS network, make sure that the correct APN is entered in Network Profile 5. Make sure that there is no space around the APN settings.

3. After turning on the phone, don't make any GSM call till the E2E check is complete. If there is an incoming GSM call before the check is complete, accept it.

4. GPSUpdateInterval vs. LocationSendInterval
   GPSUpdateInterval is the interval over which the application regularly gets the location data from the GPS chipset and caches it. This is used when sending periodic reports of GPS location. So GPSUpdateInterval should be less than LocationSendInterval. The GPS chipset gets update from the satellites every 2 seconds. The default value of GPSUpdateInterval is 120 seconds. If the LocationSendInterval is more than 120 seconds, the default value of GPSUpdateInterval is good enough. It need not be changed. If the LocationSendInterval is set to a value lower than 120 seconds, then GPSUpdateInterval has to be brought down also. Both can be set to the same value in that case. The minimum recommended value of LocationSendInterval is 60 seconds.

5. Usage of HDOP value
   The HDOP value is sent to the server along with every GPS location value. The HDOP indicates the accuracy of the GPS location value. HDOP value of > 2.0 indicates low accuracy. Low accuracy is the result of the whole sky not being visible. The satellites used to compute the location would be clustered close to each other. The more spatially distributed that the satellites used for calculating the location are, the more accurate is the result.

6. Do not use Java in background mode
   The phone supports only one Java application at any point of time. So if a Java application is running in the background, no other Java application can be started without stopping the previous one. So, we recommend that the Java applications developed for green and amber buttons to start, do some operations and terminate immediately.

7. Response matching
   All messages from the phone to the server – both HTTP and SMS – need to be acknowledged. Otherwise the messages will get queued up and retransmitted.

8. ERC Timeout
   If the ERCTimeout value is set to a very low value, then a new attempt may be made before the network disconnects the earlier call fully. You may see "SS Error" or "Not allowed" in that case. We have found that 30 seconds is a good value. "SS Error" and "Not allowed" show up occasionally. The next ERC number will be tried after 30 seconds. If you set it lower than that, you may find all call attempts failing.

9. OTA SMS
   If the incoming SMS request is longer than about 1000 bytes in case of CP 1.1 format, phone crashes have observed. In case of plain SMS, the maximum length of a single SMS is about 160,

beyond which the request is not processed. Recommendation is to change one parameter in one OTA SMS.

10. Recommendations to get most out of the battery

    a. Use tracking only during hours of day when its required. Please make use of automatic start & stop of tracking / mandown features
    b. Use tracking intervals of 3 minutes or greater
    c. Use update intervals of 3 minutes or greater
    d. User Cell-ID as location source if the user is indoors always, where there is GPS coverage
    e. Keep the LCD brightness to a minimum as possible while being indoors
    f. Avoid using hands-free / speaker mode whenever possible
    g. Charge the battery nightly
    h. Replace the battery every one year

11. getconfig
    There is a limitation in the size of HTTP POST messages that can be sent to the server. So if too many parameters are requested then the XML sent in the POST message would be truncated. Pull only up to about three or four parameters in one getconfig request.

12. Java applications that use the Camera and Audio recorder have interaction issues with loneworker application. So please avoid using those features when developing green and amber button applications.

13. When using automatic mode change using MandownStartTime, MandownStopTime, TrackingStartTime and TrackingStopTime, please make sure not to set the same time for multiple parameters. The text displayed to inform the user will be unpredictable in that case. Also, if you are setting all four values, they should be nested for the intended result. Otherwise the previous mode will be not as intended and this can cause unexpected behavior.

14. If numeric key 5 is used to send a code to the server, the torch also gets turned on when it is pressed and held. When it is pressed and held again to turn off the torch, another message is sent to the server. So it is recommended not to use key 5 for user generated key events.