XP3340 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	Comments
PB27	01-28-2011	Gomu	XP3 Lone Worker Developer Guide
PA1-PA10	06-01-2011	Gomu	Added parameters: WhiteListedNumbers, LocationSendModeAlarm, NoHandshakeForSMS, ChargerStatusReporting, UseUserSettings, LoneWorkerAccount, PreviousMode, LastKnownLocation, DelayStartupMessage, SmsSendDelay Added options: Neighbouring cell information option for LocationSource Added OTA methods: installjava, launchjava, downloadfile, downloadconfig, nbrcellinfo, callphone Added reports and responses: nbrcellinfo Added src bit value in health: Charger connected Added MSISDN parameter to OTA Server Alert Only 5 and 7 can be allowed values of ChargerStatusReporting. DownloadFile and DownloadConfig requests can take uid and pin in the Url. Added ERC call cycle timing diagram in page 18 Removed CP1.1 messages support SMS message to set APN has changed Removed Best Practices as they are already in the release notes. Added and tuned ranges for configuration parameter values.
PA11-PA14	06-27-2011	Gomu	Incorporated feedback comments
PA12	07-06-2011	Gomu	Added parameters: SMSAckTimeout, SMSMsgQueueSendInterval, LockKeypadDuringAlarm, PriorityToERCCall, AutoAcceptERCAlways, AutoAcceptWhitelistAlways, AcceptOTAFrom, AcceptConfigOTAFrom, AcceptOTAFromActions, AcceptConfigOTAFromActions Added OTA method: cancelalarm Added new reporting mode: Instant Call
PA13-PA17	11-04-2011	Gomu	Removed parameters: LockKeypadDuringAlarm Added parameters: DisableShutdownBeyond, FirstCallDelay, DisableKeypadInERCCall Added OTA method: shutdown Allowing call waiting on non-ERC calls

CONTENTS

1 Introduction

The Sonim XP3340 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 XP3300 Force to sustain even the toughest work environments. It conforms to IP68, 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, XP3340 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 XP3340 Sentinel Lone Worker functionality and its interaction with the server system.

The information in this document applies to the version of the application that shows **9.1.4.35** when Lone Worker starts up.

2 Acronyms

General Terms

Acronym	**Definition**
Client	Lone Worker client
Server	Lone Worker server
ERC	Emergency response center, Alarm receiving centre or Central Station
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
Geofence	A geographically defined boundary with reporting configured based on entry and exit or presence within or without the boundary.

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, whichever of Tilt, Idle, Fall and Impact sensors have been switched on 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.
Instant call	Alarm is sent to the server, local alert is shown to the user and a GSM call is made to the ERC.

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. *(Please note that these Java applications should not persist or run in the background as this may cause unexpected results. Typical usage would be to change the mode of the device, or sending a specific message to the server and/or making a phonecall)*

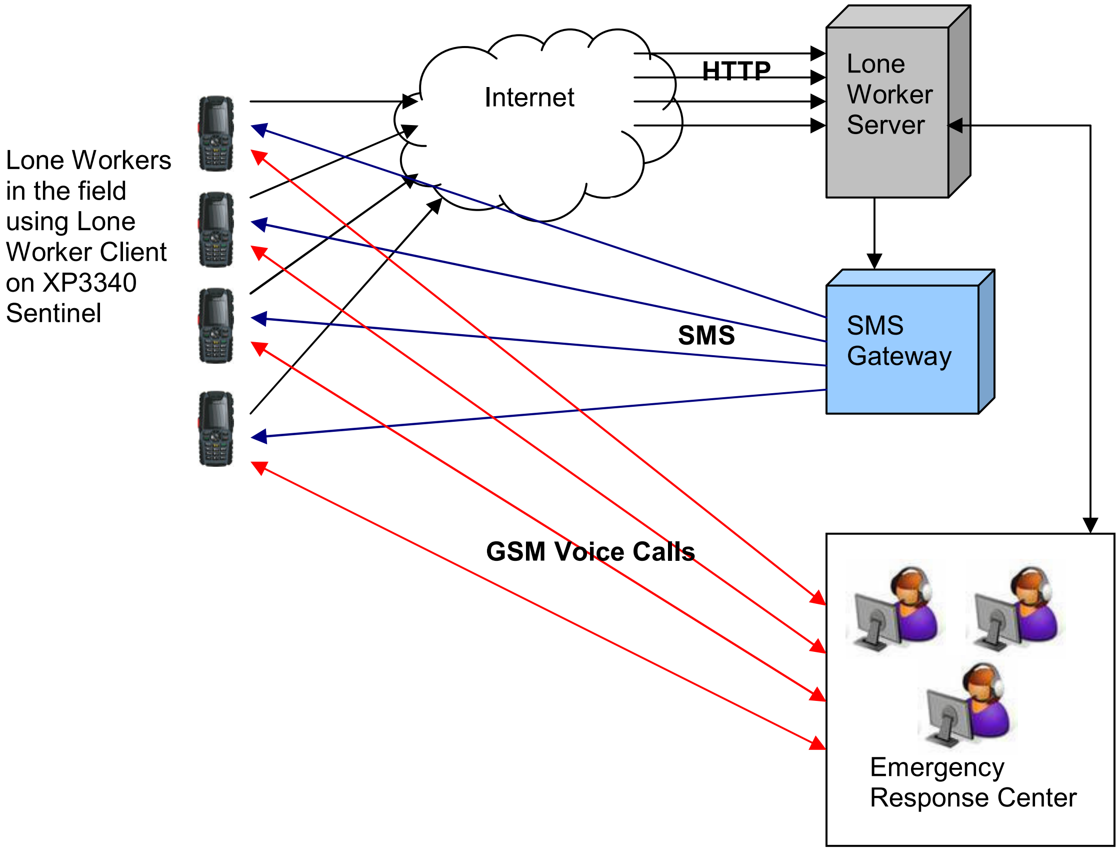
- Audio alerts should be used to alert the user about different situations. All the audio files should be configurable.

- Geofences should be configurable to generate alerts during entry/exit and presence inside/outside the areas.

- 911, 112 and 999 are recognized as emergency numbers for normal GSM call.

4 Solution Architecture

The complete Lone Worker System consists of the XP3340 Sentinel, a partner server and an Emergency Response Center (ERC).


Lone Workers in the field using Lone Worker Client on XP3340 Sentinel
Internet
HTTP
Lone Worker Server
SMS
SMS Gateway
GSM Voice Calls
Emergency Response Center

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 XP3340 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 XP3340 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 travelling. 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. *(Preferred for remote management of devices sold at Retail or where bulk provisioning is required)*

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 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 Travelling 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 Configuration lock

As we don’t want any tampering with the configuration from the phone, we lock the configuration from the server. Henceforth, the only way to change the configuration is from the server.

5.9 Changing the mode

The user can set the client to one of the several modes: disabled, simple, tracking and man down. The server and Java applications can change the mode. 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.

5.10 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.

5.11 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.

5.12 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.

5.13 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.

5.14 Travelling worker in poor network coverage

A travelling 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.

5.15 Travelling worker in trouble

A travelling 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.

5.16 Travelling worker in trouble in poor network coverage

A travelling 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.

5.17 Travelling worker in trouble in no network coverage

A travelling 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.

5.18 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.

5.19 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.

5.20 Making sure a lone worker is within a boundary

If a lone worker is supposed to be only within a geographically defined boundary, a geofence can be defined for that area. The phone can be configured to send an alert to the server if the user is outside the boundary.

5.21 Automatic clock-in and clock-out

In a place like construction site, a geofence can be defined for the worksite. Reports can be configured such that whenever the worker enters the area the server is informed of the entry and whenever the worker leaves the area the server is informed of the exit. This can be used for automatic clock-in and clock-out.

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 9999 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				
		OTA SMS requests	GPS, Battery, Awake, E2E, Health checks, manual alarm	Location tracking	Geofencing	Man down
(no icon)	Disabled (0)	Yes	No	No	No	No
◉	Simple (1)	Yes	Yes	No	No	No
⌖	Tracking (2)	Yes	Yes	Yes	Yes	No
[🚹]	Man down (3)	Yes	Yes	Yes	Yes	Yes
⚡	Alarm (4)	Yes	Yes	Yes	Depends	Depends

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

Long press of the red button will raise manual alarm. This will work even when the keypad is locked and from all screens. 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.

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. CLIP is enabled. 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.
8	Instant call	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

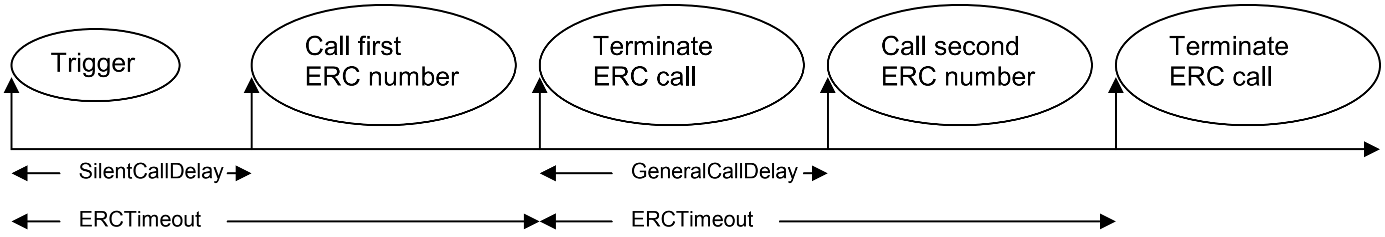
		ERC number. If not reachable, try the next number. This goes on in a loop till one of them is reached or the user cancels the alarm. The mode is changed to alarm mode. The call is put into speaker mode or normal mode depending on the configuration. CLIP is enabled.

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.

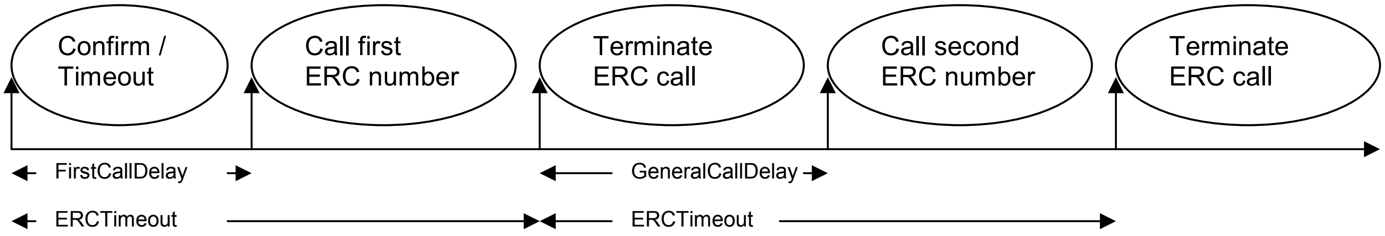
The type of alarm that is raised for different triggers mentioned in this document is the typical configuration. They can be changed.

The ERC call cycle is governed by the configuration parameters.

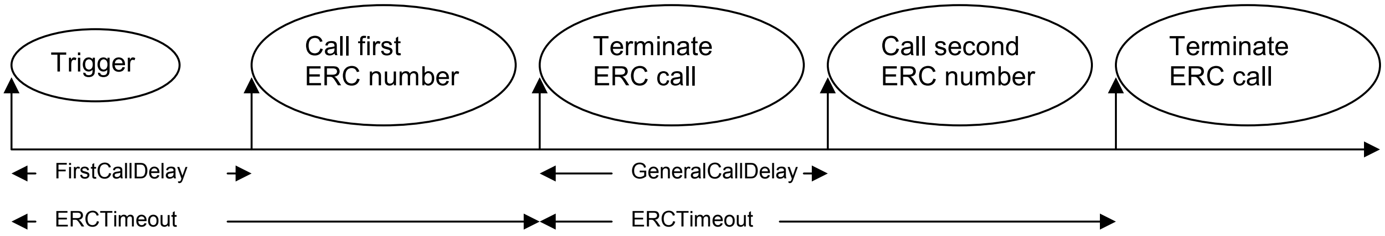
Silent alarm mode:



Alarm with call mode:



Instant call mode:



The SilentCallDelay or FirstCallDelay depending on the mode, is preempted when the response to the alarm message is got.

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. Awake check is disabled in alarm mode.

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 three seconds, *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 and 9 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. If the location source is GPS, the GPS location is sent. If the location source is Cell id, the Cell id details are sent.

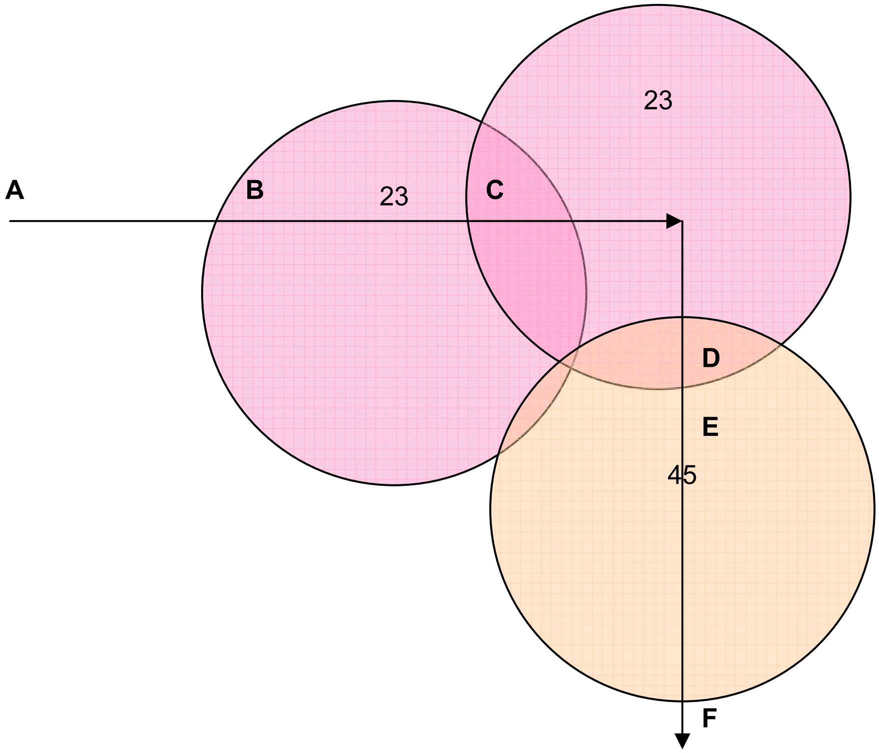
6.14 Man Down

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.

Man down check can be disabled when in alarm mode because of a previous man down event. Man down check can be disabled when the phone is connected to USB cable or wall charger. If automatic cancellation of man down is set, then tilt or idle alarm can be cancelled during pre-alert by making the phone vertical or shaking the phone, respectively.

6.15 Geofence

A set of geofences, which can be circular or rectangular can be defined. Each geofence has an id. The id can be a number between 1 and 99. Alerts can be triggered for entry or exit of any geofence.



Assume three geofences are defined with ids 23, 23 and 45 as in the picture. Suppose the phone moves from A to B to C to D to E to F following the path indicated.

- When the phone reaches point B, an entry trigger with id 23 is generated.
- When the phone reaches point C, no trigger is generated because the ids of the old and new geofences are the same.
- When the phone reaches point D, an entry trigger with id 45 is generated.
- When the phone reaches point E, an exit trigger with id 23 is generated.
- When the phone reaches point F, an exit trigger with id 45 is generated.

For rectangular geofences, the geofence specification is a set of 6 values – id, type, lat1, lon1, lat2, lon2. “type” is 0 for rectangular geofences. The lat1 and lon1 are considered as the SW vertex and lat2 and lon2 is considered as the NE vertex.

For circular geofences, the geofence specification is a set of 5 values – id, type, lat of center, lon of center and radius in meters. “type” is 1 for circular geofences.

Latitude and longitude should be in [-]ddd.dddddd format. Leading and trailing zeros need not be mentioned. For example, -12.3959 is a valid value.

The geofences are checked at the interval defined by GeofenceCheckInterval.

If the phone is in the new geofence for GeofenceThreshold (default 3) consecutive intervals, then the trigger is generated. This will prevent false alerts due to GPS offshoots.

The maximum expected accuracy is 30 meters for a geofence of 1000 meter radius. So if the phone is within 30 meters from the geofence boundary, the outcome is not guaranteed. The accuracy may be less for larger geofences.

The reporting methods for the entry and exit triggers can be configured. The geofence id is put in an info tag after the minute tag in the health messages if the reporting mode is 5 or 6. A minus in front of the id indicates exit. Otherwise it indicates entry.

The alert is triggered only if it is within the start and stop time. The format for time is hhmm. For example 1430 indicates 2:30pm local time on the phone. If the start and stop time are same, then it is triggered always.

The count specifies the number of alerts triggered when the phone is in the geofence marked for entry alert or out of the geofence marked for exit alert. If the count is 99, then the alert is generated indefinitely. Alerts are generated at every GeofenceCheckInterval.

6.16 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.

6.17 Configuration

The XP3340 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 parameters from the server. The server can request the client for one or more parameters. The number of parameters depends on the size of the name and value. Typically it can be up to three.

If a special parameter to lock the configuration is set, the SD card and the public flash memory are not read. The special parameter to lock the configuration can be set or unset only from the server.

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. The phone has to be restarted whenever the encryption key is changed.

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

6.18 GSM Call Interaction

If the parameter PriorityToGSMCall is not set, GSM calls have lower priority than alarms on Lone Worker. This is the default case.

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”, “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, no new alarm is raised. If the Lone Worker needs to show a MMI when the phone is in a call with non-ERC number, the call is terminated.

When Lone Worker MMI for any configuration is being shown, if there is an incoming GSM call, the Lone Worker MMI is closed. When Lone Worker MMI for “Local warning”, “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 and in a call with ERC, if there is an incoming GSM call from a number that is not one of the ERC numbers configured, the incoming GSM call is terminated.

If the parameter PriorityToGSMCall is set, then GSM calls take higher priority than Local Warning, Local Alarm and Server Alert. In this case, if the Lone Worker needs to show one of these MMIs when the phone is in a call with non-ERC number, the MMI is not shown. Similarly, if one of these MMIs is being shown and there is an incoming call from a non-ERC number, the MMI is closed.

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.

Existing ➔	**ERC call**	**Non-ERC call**	**LW UI**
Incoming ERC call	Terminate incoming	See table below	Depends on PriorityToERCCall
Outgoing ERC call	Terminate existing	See table below	Not applicable
Incoming non-ERC call	Terminate incoming	See table below	Depends on PriorityToGSMCall
Outgoing non-ERC call	Terminate outgoing	See table below	Not applicable
LW Local warning UI	Suppress new UI	Depends on PriorityToGSMCall	Suppress new UI
LW Local alarm UI	Suppress new UI	Depends on PriorityToGSMCall	Suppress new UI if existing >= local alarm
LW Remote alarm UI	Depends on PriorityToERCCall	End active call	Suppress new UI if existing >= remote alarm
LW Remote alarm with call UI	Depends on PriorityToERCCall	End active call	Suppress new UI if existing >= remote alarm with call
LW Instant call	Depends on PriorityToERCCall	End active call	Close UI
LW Silent call	No new call	End active call	Close UI
Miscellaneous UI on phone	Temporary popup	Temporary popup	Temporary popup

Call interaction follows the following table:

Existing ➔	**One ERC call**	**One Non-ERC call**	**Two Non-ERC call**
Incoming ERC call	Terminate incoming	Terminate existing	End held call
Outgoing ERC call	Terminate existing	Terminate existing	End active call
Incoming non-ERC call	Terminate incoming	Ask user	Ask user
Outgoing non-ERC call	Terminate outgoing	Put existing on hold	Ask user

6.19 Java Interaction

The number keys 2 and 3 are colored amber and green respectively on XP3340 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 XP3340 Sentinel. *(If you do not have access to this portal, please contact your commercial account manager who can then request access)*

The Java platform on XP3340 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.

6.20 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.

6.21 Miscellaneous Interaction

The phone supports features like launching Java application, installing Java application, etc through OTA SMS. However, whether the operation is possible or not depends on the current state of the phone. This table gives what is allowed.

	In Call	In WAP	In FG Java	In LW Alert	In BG Java
Launch Java	Not allowed	Not allowed	Not allowed	Not allowed	Not allowed
Install Java	Not allowed	Not allowed	Not allowed	Not allowed	Not allowed
Download Config File	Not allowed	Allowed	Allowed	Not allowed	Allowed
Download File	Not allowed	Allowed	Allowed	Allowed	Allowed

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 “E:”. So the Audio folder on SD card memory will be “E:\Audio”. The Bluetooth transfer folder is accessible at the path “C:”. So the Audio folder on the Bluetooth transfer folder will be “C:\Audio”.

Parameter	Type	Description	Client configurable	Default	Example
UserId	Uint32	Identification of the user. Range – 1-9999	No		2093
SystemName	String	Name of the lone worker system. Max 20 chars	No		Akela Hoon
UserName	String	Name of the user. Max 20 chars	No		Agent 007
EncryptionKey	String	Encryption key. 5 to 10 chars	No	disabled	
DataVersionPublic	Unit16	Version number of data taken from public flash area. Range – 0-99	No	0	
DataVersionSD	Uint16	Version number of data taken from SD card. Range – 0-99	No	0	
ConfigurationLocked	Boolean	Is the configuration locked against local changes?	No	0	
AcceptOTAFrom	String	Comma separated list of MSISDN numbers from which OTA query SMS are allowed. Maximum of 5 numbers are allowed.	No		+919844198441, +919844198442, +919844198443
AcceptConfigOTAFrom	String	Comma separated list of MSISDN	No		+919844198441, +919844198442,

		numbers from which OTA configuration SMS are allowed. Maximum of 5 numbers are allowed.			+919844198443
AcceptOTAFromActions	String	OTA query actions	No		location,cellid,nbrcellinfo,health,getconfig,alert,vibrate
AcceptConfigOTAFromActions	String	OTA configuration actions	No		changeconfig,restart,raiseevent,addtophonebook,installjava,launchjava,downloadfile,downloadconfig,setapn,cancelalarm
HttpUrl	String	Http url to send all data to. Max 40 chars	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 3 => None	Yes	2	
LoneWorkerAccount	String	Identifier of the network APN settings to be used	No		Network Profile 5
QueueSize	Uint16	Total number of messages to queue if out of coverage. Range – 50-300.	No	300	
Queue50Reporting	Values	One of the reporting modes to report 50% queue full	Yes	0	
Queue80Reporting	Values	One of the reporting modes to	Yes	1	

		report 80% queue full			
Queue100Reporting	Values	One of the reporting modes to report 100% queue full	Yes	1	
DisableWap	Boolean	Should WAP be disabled	Yes	0	
NoHandshakeForSMS	Boolean	Should handshake be disabled for SMS transport	Yes	0	
SmsSendDelay	Uint16	Number of seconds to delay sending response to request to allow SMS subsystem to settle down. Range – 0-5	Yes	0	
SMSAckTimeout	Uint16	Number of seconds to wait for the response to an SMS before resending it. Range – 30-120	Yes	60	
SMSMsgQueueSendInterval	Uint16	Number of seconds to wait between two SMS sends	Yes	10	
LocalWarningDuration	Uint16	Number of seconds to ring local warning before taking the next action. Should be less than the Phone settings > Power Saving > LCD backlight > Time setting. Range – 2-60	Yes	5	
LocalAlarmDuration	Uint16	Number of seconds to ring local alarm before taking the next action. Should be less than the Phone settings > Power Saving >	Yes	10	

		LCD backlight > Time setting. Range – 2-60			
LongBeepInterval	Uint16	Number of seconds duration and interval before repeating local alarm. Minimum 3, If this is shorter than the audio message, it will be truncated and repeated. If this is longer than the audio message, there will be a gap before repeating.	Yes	3	
ShortBeepInterval	Uint16	Number of seconds interval before repeating local beep after activating the alarm. Minimum 3, If this is shorter than the audio message, it will be truncated and repeated. If this is longer than the audio message, there will be a gap before repeating.	Yes	10	
LongBeepAudioMessage	String	Path to the audio file containing the audio message to be played repeatedly before activating the alarm. Max 40 chars	Yes		
ShortBeepAudioMessage	String	Path to the audio file containing the audio message to be played repeatedly after activating the alarm. Max 40 chars	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	
DelayStartupMessage	Uint16	Number of seconds to delay sending messages to wait for GPRS to be available. Range – 0-120	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
ERCNumbersForSilentCall	String	If provided, this will override the ERCNumbers for silent call. 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. Range – 20-60. This should be at least 10 more than FirstCallDelay, SilentCallDelay and GeneralCallDelay.	Yes	30	
WhiteListedNumbers	String	Comma separated list of MSISDN numbers which have to be auto-answered. Maximum of 5 numbers are	Yes		+919844198441, +919844198442, +919844198443

		allowed.			
UseSpeaker	Boolean	Should speaker be used for calls from and to ERC?	Yes	1	
EnableCLIP	Boolean	Should CLIP be enabled before making call to ERC?	Yes	0	
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. Range – 10-120	Yes	30	
SilentCallDelay	Uint16	Maximum number of seconds to delay calling ERC before starting silent call. Range – 1-10. This should be at least 10 less than ERCTimeout	Yes	5	
FirstCallDelay	Uint16	Maximum number of seconds to delay calling ERC before starting non-silent call. Range – 1-10. This should be at least 10 less than ERCTimeout	Yes	10	
GeneralCallDelay	Uint16	Number of seconds to delay calling ERC when calling again. Range – 1-10. This should be at least	Yes	10	

		10 less than ERCTimeout			
ContinueCallingERC	Boolean	Should ERC call cycle be continued after disconnecting from an answered ERC number	Yes	0	
PriorityToGSMCall	Boolean	Should non-ERC GSM call be given more priority than Local Warning, Local Alarm and Server Alert?	Yes	0	
PriorityToERCCall	Boolean	Should ERC GSM call be given more priority than UI?	Yes	1	
UseUserSettings	Boolean	Should the user profile volume settings be used for audio alarms?	Yes	1	
DisableShutdownBeyond	Values	The mode beyond which shutdown should be disabled.	Yes	5	
AutoAcceptERCAlways	Boolean	Should calls from ERC be accepted automatically when phone is not in alarm mode?	Yes	1	
AutoAcceptWhitelistAlways	Boolean	Should calls from Whitelist be accepted automatically when phone is not in alarm mode?	Yes	1	
DisableKeypadInERCCall	Boolean	Should keypad be disabled when in call with ERC?	Yes	0	
LoneWorkerMode	Values	One of the lone worker modes	Yes	0	
LoneWorkerMinMode	Values	One of the configurable lone worker modes. Should not be	Yes	1	

		alarm mode.			
PreviousMode	Values	If the current mode is alarm, this has the previous mode	No		
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 normal health messages. The emergency level of the message can increase the transport mode.	Yes	1	
HealthSendTimeout	Uint16	Time in seconds to wait for health send	Yes	60	

		response before alerting the user. Range – 30-300			
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 regular intervals	Yes	0	
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	Yes	30	

		current location. Range – 30-300			
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	
ChargerStatusReporting	Values	One of the reporting modes. Only 5 and 7 are valid values.	Yes	7	
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	
AwakeCheckMessage	String	Text to be displayed before activating the alarm. Max 40 chars	Yes		
AwakeCheckAudioMessage	String	Path to the audio file containing the audio message to be played repeatedly before activating the alarm. Max 40 chars	Yes		

AwakeCheckMessageInterval	Uint16	Interval and duration to play Audio message. Minimum 3, If this is shorter than the audio message, it will be truncated and repeated. If this is longer than the audio message, there will be a gap before repeating.	Yes	10	
TechnicalCheckAudioMessage	String	Path to the audio file containing the audio message to be played repeatedly before activating the alarm. Max 40 chars	Yes		
TechnicalCheckMessageInterval	Uint16	Interval and duration to play Audio message. Minimum 3, If this is shorter than the audio message, it will be truncated and repeated. If this is longer than the audio message, there will be a gap before repeating.	Yes	10	
E2ECheck	Boolean	If to check E2E connectivity at regular intervals	Yes	0	
E2ECheckInterval	Uint16	Interval of checking in seconds. This should be more than E2ETimeout. Range – 300-43200	Yes	600	
E2ETimeout	Uint16	Timeout value in seconds. This should be less than E2ECheckInterval Range – 10-600	Yes	60	

E2EFailureReporting	Values	One of the reporting modes	Yes	1	
E2ESuccessReporting	Values	One of the reporting modes – 0, 1, 7	Yes	0	
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	
LocationSource	Values	0 => GPS 1 => CellId 2 => Neighbouring cell information	Yes	0	
LocationSendInterval	Uint16	Interval of sending in seconds. Range – 10-43200	Yes	600	
LocationSendIntervalAlarm	Uint16	Interval of sending in seconds in alarm mode. Range – 10-43200	Yes	600	
LocationSendIntervalSecondary	Uint16	Approximate interval of sending in seconds when SendVia is 2, LocationSendMode is 3, not in alarm mode and the primary transport is not available. Range – 0 or 10-43200	Yes	0	
LocationSendMode	Values	Transport mode for sending location	Yes	2	
LocationSendModeAlarm	Values	Transport mode for sending location in alarm state	Yes	3	

LastKnownLocation	String	GPS values for the last known location	No		12.911795,77.599625,1,913.80,0.15,96.58,20110512130852
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. Range – 5-20	Yes	5	
MandownIdleTimeout	Uint16	Maximum duration in seconds for which the phone can be without movement. Range – 10-300	Yes	20	
MandownTiltThreshold	Uint16	Tilt threshold in degrees beyond which the phone is considered tilted. Range – 5-75	Yes	20	
MandownTiltTimeout	Unit16	Maximum duration in seconds for which the phone can be tilted. Range – 10-3600	Yes	10	
MandownImpactThreshold	Uint16	Acceleration threshold in percentage of 8G beyond which alarm should be raised. Range – 75-95	Yes	80	
MandownImpactTimeout	Unit16	Maximum duration in milliseconds for	Yes	10	

		which the phone should be beyond acceleration threshold. Range – 5-500			
MandownFallThreshold	Uint16	Acceleration threshold in percentage of 8G below which alarm should be raised. Range – 3-10	Yes	7	
MandownFallTimeout	Unit16	Minimum duration in milliseconds for which the phone should fall. Range – 200-500 Approximately, 200 => 12-18 inches 250 => 18-24 inches 350 => 30-36 inches 500 => 60-66 inches	Yes	200	
MandownTextMessage	String	Text to be displayed before activating the alarm. Max 40 chars	Yes		
MandownAudioMessage	String	Path to the audio file containing the audio message to be played repeatedly before activating the alarm. Max 40 chars	Yes		
MandownMessageInterval	Uint16	Interval and duration to play Audio message. Minimum 3, If this is shorter than the audio message, it will be truncated and repeated. If this is longer than the audio message, there will be a gap before repeating.	Yes	10	

MandownReporting	Values	One of the reporting modes	Yes	3	
DisableMandownAfterAlarm	Boolean	If mandown should be disabled after getting into alarm mode because of a mandown event	Yes	0	
DisableMandownWhenCharging	Boolean	If mandown should be disabled when wall charger or USB cable is connected	Yes	0	
MandownAutomaticCancel	Boolean	If the pre-alert for Tilt and Idle alarms should be cancelled when the situation is no longer present	Yes	0	
ManualAlarm	Boolean	If to support manual alarm	Yes	1	
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 4, 5, 6, 7, 8, 9	Yes	0	
NumberKeyAlarmReporting	Values	One of the reporting modes	Yes	5	
EnableGeofence	Boolean	If to check geofence	Yes	No	
Geofence0	String	Comma separated	Yes		**45**, 0, 12.90,

		list – geofence specification, entry reporting mode, exit reporting mode, start time, end time, count			79.80, 12.91, 80, 5, 5, 800, 1700, 2
Geofence1	String	Comma separated list – geofence specification, (entry reporting mode, exit reporting mode, start time, end time, count)	Yes		**45**, 0, 12.90, 79.80, 12.91, 80
Geofence2	String	Comma separated list – geofence specification, (entry reporting mode, exit reporting mode, start time, end time, count)	Yes		
Geofence3	String	Comma separated list – geofence specification, (entry reporting mode, exit reporting mode, start time, end time, count)	Yes		
Geofence4	String	Comma separated list – geofence specification, (entry reporting mode, exit reporting mode, start time, end time, count)	Yes		
Geofence5	String	Comma separated list – geofence specification, (entry reporting mode, exit reporting mode, start time, end time, count)	Yes		
Geofence6	String	Comma separated list – geofence specification, (entry reporting mode, exit reporting mode, start time,	Yes		

		end time, count)			
Geofence7	String	Comma separated list – geofence specification, (entry reporting mode, exit reporting mode, start time, end time, count)	Yes		
Geofence8	String	Comma separated list – geofence specification, (entry reporting mode, exit reporting mode, start time, end time, count)	Yes		
Geofence9	String	Comma separated list – geofence specification, (entry reporting mode, exit reporting mode, start time, end time, count)	Yes		
SendInfoInHealth	Boolean	If to send info tag in health/alarm messages	Yes	No	
GeofenceCheckInterval	Uint16	Interval of checking GPS location in seconds. Range – 2-43200	Yes	600	
GeofenceThreshold	Uint16	Number of times that the phone should be inside or outside of the geofence before raising alert. Minimum 1	Yes	3	
GeofenceAccuracy	Values	Accuracy for a geofence of 1km radius. The accuracy will decrease in proportion to the radius. 0 – 130m 1 – 60m 2 – 30m	Yes	2	

GeofenceReportAtStart	Boolean	If to raise the entry and exit alerts based on current position when turning on the phone	Yes	No	

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 120 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.

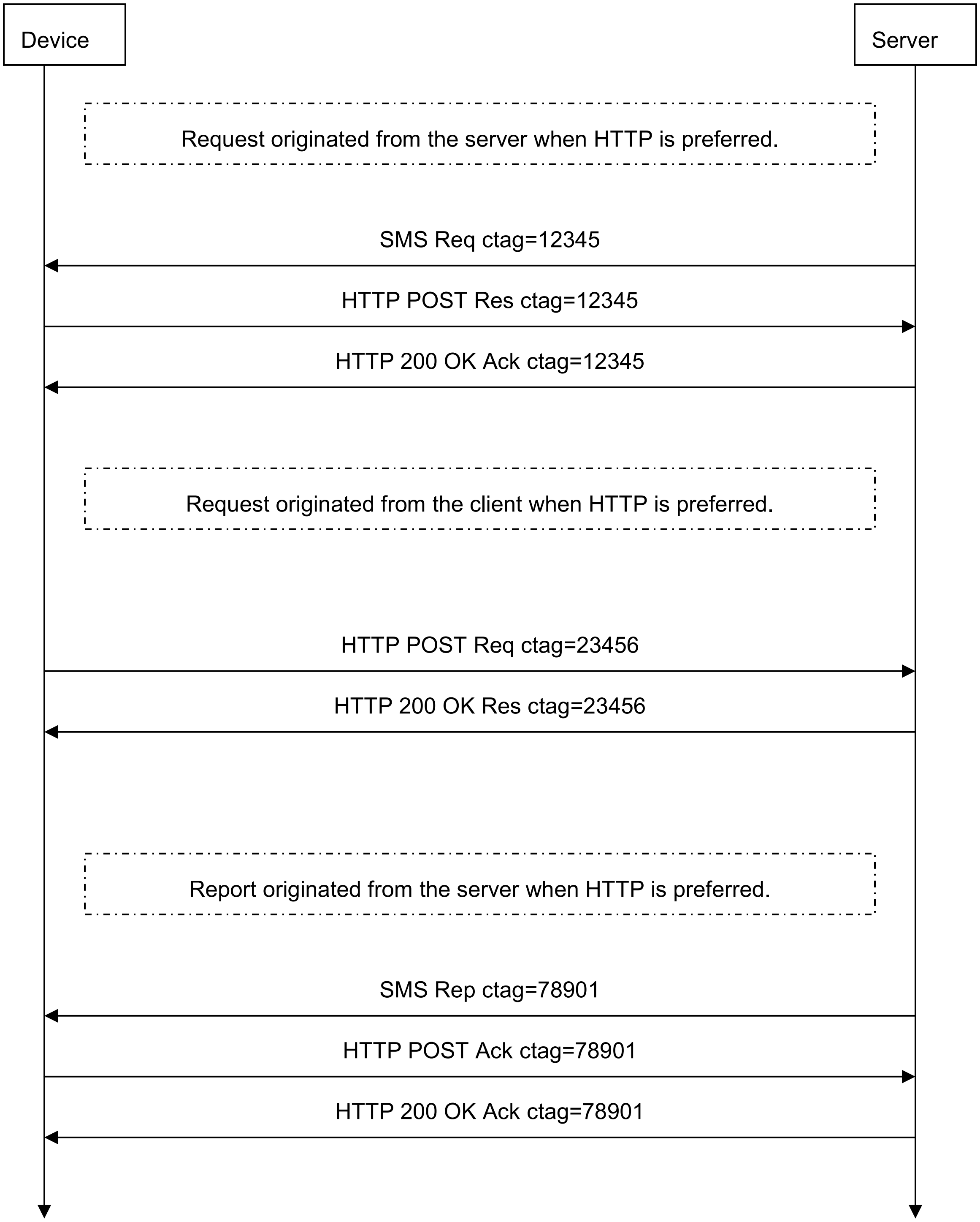
- IMSI is not sent in the messages.
- The first 5 digits of the IMEI is dropped.
- Accelerometer and Cell id values are not sent in health report and response.

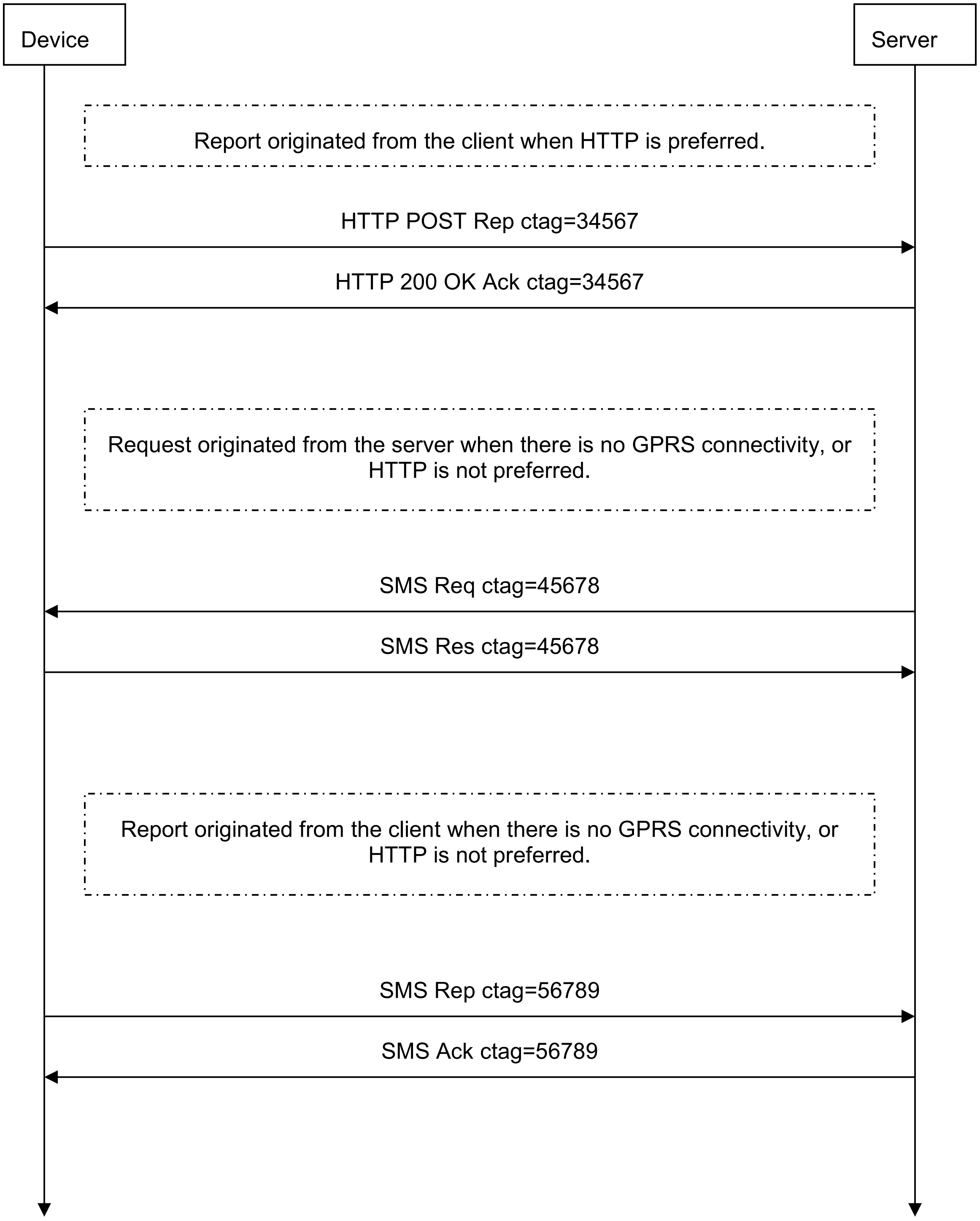
The order of the tags is fixed. Please see the examples.

<ctag>, <uid> and <minutes> fields should not be zero.

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
Request originated from the server when HTTP is preferred.
SMS Req ctag=12345
HTTP POST Res ctag=12345
HTTP 200 OK Ack ctag=12345
Request originated from the client when HTTP is preferred.
HTTP POST Req ctag=23456
HTTP 200 OK Res ctag=23456
Report originated from the server when HTTP is preferred.
SMS Rep ctag=78901
HTTP POST Ack ctag=78901
HTTP 200 OK Ack ctag=78901


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 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.

8.1.11 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
```

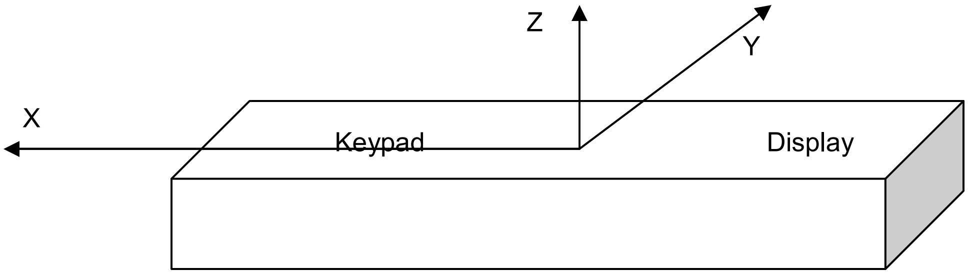
The values in **bold** have to be replaced by the correct values. Instead of |, # or comma also can be used.

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

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

8.1.12 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>last</mode>
<timeout>20</timeout>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
REQ#location#20090#1#last#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#
```

8.2.9 NbrCellInfo Request

No parameters. Send the details of the servicing cell and up to six neighbouring cells.

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>nbrcellinfo</action>
<ctag>20090</ctag>
<uid>1</uid>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
REQ#nbrcellinfo#20090#1#345#
```

8.2.10 NbrCellInfo 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
Arfcn	ARFCN of the current cell	Mandatory	23
Bsic	BSIC of the current cell	Mandatory	45
Rxlev	Rx level of the current cell	Mandatory	56
Ta	Timing advance of the current cell	Mandatory	1
Numnbr	Number of neighbouring cells	Mandatory	4
…	Mcc, Mnc, Lac, Cellid, Arfcn, Bsic and Rxlev of each neighbouring cell follows		

If there is no GSM coverage currently, send 404 as the status.

Here is an example:

```
<? xml version="1.0" encoding="utf-8" ?>
<res>
<action>nbrcellinfo</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>
<arfcn>23</arfcn>
<bsic>45</bsic>
<rxlev>56</rxlev>
<ta>1</ta>
<numnbr>2</numnbr>
<mcc>234</mcc>
<mnc>10</mnc>
<lac>14334</lac>
<cellid>21400</cellid>
<arfcn>24</arfcn>
<bsic>42</bsic>
<rxlev>26</rxlev>
<mcc>234</mcc>
<mnc>10</mnc>
<lac>12345</lac>
<cellid>21310</cellid>
<arfcn>24</arfcn>
<bsic>41</bsic>
<rxlev>36</rxlev>
<minutes>345</minutes>
</res>
```

Here is the text form of the same message:

```
RES#nbrcellinfo#20090#78965#123456789012345#123456789012345#200#234#10#14345#2
1410#23#45#56#1#2#234#10#14334#21400#24#42#26#234#10#12345#21310#24#41#36#345#
```

8.2.11 NbrCellInfo Report

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
Arfcn	ARFCN of the current cell	Mandatory	23
Bsic	BSIC of the current cell	Mandatory	45
Rxlev	Rx level of the current cell	Mandatory	56
Ta	Timing advance of the current cell	Mandatory	1
Numnbr	Number of neighbouring cells	Mandatory	4
…	Mcc, Mnc, Lac, Cellid, Arfcn, Bsic and Rxlev of each neighbouring cell follows		

If there is no GSM coverage currently, send 404 as the status.

Here is an example:

```
<? xml version="1.0" encoding="utf-8" ?>
<rep>
<action>nbrcellinfo</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>
<arfcn>23</arfcn>
<bsic>45</bsic>
<rxlev>56</rxlev>
<ta>1</ta>
<numnbr>2</numnbr>
<mcc>234</mcc>
<mnc>10</mnc>
<lac>14334</lac>
<cellid>21400</cellid>
<arfcn>24</arfcn>
<bsic>42</bsic>
<rxlev>26</rxlev>
<mcc>234</mcc>
<mnc>10</mnc>
<lac>12345</lac>
<cellid>21310</cellid>
<arfcn>24</arfcn>
<bsic>41</bsic>
<rxlev>36</rxlev>
<minutes>345</minutes>
</rep>
```

Here is the text form of the same message:

```
REP#nbrcellinfo#20090#78965#123456789012345#123456789012345#200#234#10#14345#2
1410#23#45#56#1#2#234#10#14334#21400#24#42#26#234#10#12345#21310#24#41#36#345#
```

8.2.12 NbrCellInfo 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 (not used) 12 => Mode change (not used) 13 => Amber key (number key 2) pressed 14 => Green key (number key 3) pressed 15 => Number key 0 pressed (not used)	Mandatory	b000000001

	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 23 => Geofence entry 24 => Geofence exit 25 => Charger connected		
Mode	Lone worker mode. Previous mode is filled here if current mode is alarm.	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	Mandatory	b000000001

	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 (not used) 12 => Mode change (not used) 13 => Amber key (number key 2) pressed 14 => Green key (number key 3) pressed 15 => Number key 0 pressed (not used) 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 23 => Geofence entry 24 => Geofence exit 25 => Charger connected		
Mode	Lone worker mode. Previous mode is filled here if current mode is alarm.	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 Cancel Alarm

8.4.1 Cancel Alarm Request

No parameters. If the current mode is alarm, switch to the previous mode before alarm.

This is sent from the server to the mobile.

Here is an example.

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

Here is the text form of the same message:

```
REQ#cancelalarm#20090#1#345#
```

8.5 Configuration

8.5.1 GetConfiguration Request

Tag	Description	Occurrence	Example
Params	Comma separated list of parameters to get	Mandatory	UserId,LoneWorkerMode

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#
```

8.5.2 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#
```

8.5.3 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#
```

8.5.4 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#
```

8.6 Alert

8.6.1 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	OK
Sk2	Soft key option 2		Cancel
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
Msisdn	Msisdn to call if Sk1 is selected. Applicable only when both option keys are used.	Optional	+919844198441

A MMI screen is shown to the user. Alarm is raised if specified. User response is sent back. If only one soft key option is given, it is shown on the right. If both are given, Sk1 is shown on left and Sk2 is shown on right. The options can be on of these texts: OK, On, Option, Pause, Off, Reject, Reply, Exit, Call, Back, Cancel, Yes, No, Close. If an MSISDN number is provided (non empty), then a GSM call is made to the number if Sk1 is pressed.

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>Yes</sk1>
<sk2>No</sk2>
<timeout>20</timeout>
<alarm>Y</alarm>
<duration>10000</duration>
<msisdn>+919844198441</msisdn>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
REQ#alert#20090#1#Attention#Do you need
help?#Yes#No#20#Y#10000#+919844198441#345#
```

8.6.2 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#345#
```

8.7 Restart

8.7.1 Restart Request

No parameters. Restart the phone.

This is sent from the server to the mobile.

Here is an example.

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

Here is the text form of the same message:

```
REQ#restart#20090#1#345#
```

8.8 Vibrate

8.8.1 Vibrate Request

No parameters. Vibrate the phone for one second.

This is sent from the server to the mobile.

Here is an example.

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

Here is the text form of the same message:

```
REQ#vibrate#20090#1#345#
```

8.9 Raise Event

8.9.1 Raise event Request

Tag	Description	Occurrence	Example
Mode	Reporting mode	Mandatory	1
Title	Title to be displayed if applicable	Mandatory	Attention
Message	Message to be displayed if applicable	Mandatory	Meeting cancelled

Initiate an event with “Server request” as the incident and the specified reporting mode. Title and Message are applicable for reporting modes 0, 1, 2, 3.

This is sent from the server to the mobile.

Here is an example.

```
<? xml version="1.0" encoding="utf-8" ?>
<req>
<action>raiseevent</action>
<ctag>20090</ctag>
<uid>1</uid>
<mode>1</mode>
<title>Attention</title>
<message>Meeting cancelled</message>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
REQ#raiseevent#20090#1#1#Attention#Meeting cancelled#345#
```

8.10 Call Phone

8.10.1 Call phone Request

Tag	Description	Occurrence	Example
MSISDN	MSISDN of the phone to call	Mandatory	+919844198441

Initiate a GSM call with the given number.

This is sent from the server to the mobile.

Here is an example.

```
<? xml version="1.0" encoding="utf-8" ?>
<req>
<action>callphone</action>
<ctag>20090</ctag>
<uid>1</uid>
<msisdn>+919844198441</msisdn>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
REQ#callphone#20090#1#+919844198441#345#
```

8.10.2 Call phone Response

Tag	Description	Occurrence	Example
Status	200 – Success 400 – Syntax error 503 – Phone is busy in call or LW UI is active. Try again later.	Mandatory	200

This is sent from the mobile to the server.

Here is an example:

```
<? xml version="1.0" encoding="utf-8" ?>
<res>
<action>callphone</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#callphone#20090#1#356789431226188#234456789796861#200#345#
```

8.11 Phone Book

8.11.1 Add to phone book Request

Tag	Description	Occurrence	Example

Name	Display name	Mandatory	Dispatcher
Number	Phone number	Mandatory	+919844198441

Adds the number to the phone book.

This is sent from the server to the mobile.

Here is an example.

```
<? xml version="1.0" encoding="utf-8" ?>
<req>
<action>addtophonebook</action>
<ctag>20090</ctag>
<uid>1</uid>
<name>Dispatcher</name>
<number>+919844198441</number>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
REQ#addtophonebook#20090#1#Dispatcher#+919844198441#345#
```

8.11.2 Add to phone book Response

Tag	**Description**	**Occurrence**	**Example**
Status	200 – Success 400 – Syntax error 500 – Error 503 – Phone is busy in call. Try again later.	Mandatory	200

This is sent from the mobile to the server.

Here is an example:

```
<? xml version="1.0" encoding="utf-8" ?>
<res>
<action>addtophonebook</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#addtophonebook#20090#1#356789431226188#234456789796861#200#345#
```

8.12 Java Application Control

8.12.1 InstallJava Request

Tag	Description	Occurrence	Example
Url	HTTP URL of the path to JAR or JAD file	Mandatory	http://host.domain.com/path/JavaApp.jar

Installs the Java application on the phone after getting confirmation from the user.

This is sent from the server to the mobile.

Here is an example.

```
<? xml version="1.0" encoding="utf-8" ?>
<req>
<action>installjava</action>
<ctag>20090</ctag>
<uid>1</uid>
<url>http://host.domain.com/path/JavaApp.jar</url>
<status>200</status>
<minutes>345</minutes>
```

Here is the text form of the same message:

```
REQ#installjava#20090#1#http://host.domain.com/path/JavaApp.jar#345#
```

8.12.2 InstallJava Response

Status	200 – Success 400 – Syntax error 500 – Error 503 – Phone is busy in call, Java/WAP is active or LW MMI is displayed. Try again later.	Mandatory	200

This is sent from the mobile to the server.

```
<? xml version="1.0" encoding="utf-8" ?>
<res>
<action>installjava</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#installjava#20090#1#356789431226188#234456789796861#200#345#
```

8.12.3 LaunchJava Request

Tag	Description	Occurrence	Example
Name	Name of the midlet	Mandatory	JavaApp

Launches the Java application on the phone if there is an Midlet of that name installed. The name is case sensitive.

This is sent from the server to the mobile.

Here is an example.

```
<? xml version="1.0" encoding="utf-8" ?>
<req>
<action>launchjava</action>
<ctag>20090</ctag>
<uid>1</uid>
<name>JavaApp</name>
<status>200</status>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
REQ#launchjava#20090#1#JavaApp#345#
```

8.12.4 Launch Java Response

Status	200 – Success 400 – Syntax error 500 – Error 503 – Phone is busy in call, Java/WAP is active or LW MMI is displayed. Try again later.	Mandatory	200

This is sent from the mobile to the server.

Here is an example:

```
<? xml version="1.0" encoding="utf-8" ?>
<res>
<action>launchjava</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#launchjava#20090#1#356789431226188#234456789796861#200#345#
```

8.13 Download File

8.13.1 DownloadFile Request

Tag	Description	Occurrence	Example
Url	HTTP URL of the path to the file	Mandatory	http://host.domain.com/path/file.jpg
Filename	Full path to local file name to be saved as	Mandatory	C:/Others/file.jpg

Downloads the URL and saves the content into the filename specified. If the Url does not include HTTP parameters uid and pin, the phone sends the current userid as the uid and 0 as the pin. The phone's imei number is sent always.

This is sent from the server to the mobile.

Here is an example.

```
<? xml version="1.0" encoding="utf-8" ?>
<req>
<action>downloadfile</action>
<ctag>20090</ctag>
<uid>1</uid>
<url>http://host.domain.com/path/file.jpg</url>
<filename>C:/Others/file.jpg</filename>
<status>200</status>
<minutes>345</minutes>
```

Here is the text form of the same message:

```
REQ#downloadfile#20090#1#http://host.domain.com/path/file.jpg#C:/Others/file.j
pg#345#
```

8.13.2 DownloadFile Response

Status	200 – Success 400 – Syntax error 404 – Not found 500 – Error 503 – Phone is busy in call, Java/WAP is active or LW MMI is displayed. Try again later.	Mandatory	200

This is sent from the mobile to the server.

Here is an example:

```
<? xml version="1.0" encoding="utf-8" ?>
<res>
<action>downloadfile</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#downloadfile#20090#1#356789431226188#234456789796861#200#345#
```

8.14 Download Configuration

8.14.1 DownloadConfig Request

Tag	Description	Occurrence	Example
Url	HTTP URL of the path to the configuration file	Mandatory	http://host.domain.com/path/LWConfig.txt

Downloads the URL and treats it as a list of lines with name=value pairs and applies the configuration directly into the Lone Worker application. If the Url does not include HTTP parameters uid and pin, the phone sends the current userid as the uid and 0 as the pin. The phone's imei number is sent always. For example, to pull user information from the demo server, a Url like this can be used:
http://125.63.77.148/server/LWS/index.php?action=GetConfig&uid=20&pin=40

This is sent from the server to the mobile.

Here is an example.

```
<? xml version="1.0" encoding="utf-8" ?>
<req>
<action>downloadconfig</action>
<ctag>20090</ctag>
<uid>1</uid>
<url>http://host.domain.com/path/LWConfig.txt</url>
<status>200</status>
<minutes>345</minutes>
</req>
```

Here is the text form of the same message:

```
REQ#downloadconfig#20090#1#http://host.domain.com/path/LWConfig.txt#345#
```

8.14.2 DownloadConfig Response

Status	200 – Success 400 – Syntax error 404 – Not found 500 – Error 503 – Phone is busy in call, Java/WAP is active or LW MMI is displayed. Try again later.	Mandatory	200

This is sent from the mobile to the server.

Here is an example:

```
<? xml version="1.0" encoding="utf-8" ?>
<res>
<action>downloadconfig</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#downloadconfig#20090#1#356789431226188#234456789796861#200#345#
```

8.15 Set SUPL Request

Tag	Description	Occurrence	Example

AGPS	AGPS can be set to On/Off	Mandatory	If On activates AGPS on mobile
Url	HTTP URL of SUPL server	Mandatory	supl.nokia.com
Port	Port number	Mandatory	7275

If AGPS is set to ON, it sends request to AGPS server and server sends back assistance data to mobile through GPRS. GPS receiver uses this data to calculate TTFF (Time to first fix).

This is sent from server to mobile.

Here is the text form of the same message:

```
REQ#setsupl#42578#54#1#supl.nokia.com#7275#462
```

8.16 Shutdown

8.16.1 Shutdown Request

No parameters. Phone is shutdown.

This is sent from the server to the mobile.

Here is an example.

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

Here is the text form of the same message:

```
REQ#shutdown#20090#1#345#
```

9 Best Practices

Please refer to the release notes for the *Best Practices* to get the best out of the Sonim Sentinel phone.