



WdpStats

USER MANUAL

Version 2.9

Frédéric SACCOCCIO

CONTENTS

1	Abstract	4
1.1	Typesetting conventions	4
2	Description	4
3	File Name	4
4	Collected information	5
4.1	Periodic.....	5
4.2	At event	7
5	Transmission of the collected informations	8
6	Schedule management	8
6.1	General considerations	8
6.2	Syntax.....	8
6.2.1	Schedule profiles.....	8
6.2.2	Schedule index.....	9
6.3	Examples	10
7	Configuration	20
7.1	Periodical statistics with or without threshold	20
7.2	Periodical statistics with or without change	20
7.3	At Event	21
7.4	Main Section.....	21
7.5	Configuration file sample:	22
8	wdp.stats Structure	24
8.1	V1.1 Lines	24
8.2	V 2.0 lines	25
8.3	V 3.0 Header.....	26
8.4	V3.0 Data header.....	26
8.5	V3.0 Data line	26
8.5.1	GPRS statistics.....	27
8.5.1.1	g1	27
8.5.1.2	g2	27
8.5.1.3	g3	27
9	Examples	28
10	List of statistics IDs for V3 output	29
10.1	General statistics	29
10.2	Wi-Fi statistics	29
10.3	WireLess Telnet statistics.....	30
10.4	GPRS statistics	30
10.5	GPS statistics	30
10.6	Hits and falls statistics.....	30
11	Platform versions of WdpStats V3	31
11.1	Wifi	31
11.2	Free fall.....	32

Document history

Date - Version	Author(s)	Modifications
2009/07/06 - Version 2.2	-	Graphs and some events added
2009/12/29- Version 2.3	-	MinAlways, MaxAlways methods added and changes in the way of interpreting Average.
2011/03/11 - Version 2.4	-	Parameters GPRS added.
2012/03/15 - Version 2.5	-	New output file version 3.0
2014/11/06 - Version 2.6	Frédéric SACCOCCIO	<ul style="list-style-type: none">▪ New statistic: Free fall event.▪ Document cleaning and minor fixes
2014/11/25 - Version 2.7		<ul style="list-style-type: none">▪ Improved the tables to summarize all the statistics and their IDs: they have been split into categories.▪ Updated the GPS statistics ID to p0 (previously GP)▪ Added the detailed format of the g3 GPRS statistics▪ Removed the Windows XP specific statistics
2014/12/22 – Version 2.8	Frédéric SACCOCCIO	<ul style="list-style-type: none">▪ Added the schedule management feature.
2015/01/19 – Version 2.9	Frédéric SACCOCCIO	<ul style="list-style-type: none">▪ Added the MOTOROLA MC32N0 in the devices that supports the “Free Fall” statistic.

1 Abstract

This document is the user manual of WdpStats: the statistics module of the Symphony client. The WdpStats module records various statistics and sends them to the Symphony Console. Once they are received on the Symphony Console, the transmitted statistics can be processed to establish reports.

This document covers the **3.1.2.0** version (and superior) of WDPStats.

1.1 Typesetting conventions

A new feature that is available from a particular versión is typeset like below:

new Version 3.2.5.0

2 Description

The WdpStats module records and sends **2** main types of statistics:

1. *Periodic:*

1.1. *With or without threshold.*

This kind of data is a periodically computed and/or retrieved quantity. A threshold can be set to decide how the statistics is considered (recorded or not).

1.2. *With or without "change".*

This kind of data is a periodically computed and/or retrieved quantity. A setting can be set to decide if the statistics is recorded only if it has changed since the last retrieval.

2. *At event.*

This kind of data is usually not a numeric one but rather an event that may occur at any time (network connection/disconnection, reset, free fall).

The statistics are logged into a file that is sent to the Symphony server by the client every time it sends a LIST or an UPDATE command.

Each type of information can be parameterized independently. It is even possible to have two different thresholds for a given type of information, for example: two thresholds for the battery: 40% and 20%.

Each recorded statistic can have its own name in order to identify it in the console.

3 File Name

On the terminal, the file name is:

wdpX.stats (where X is a number from **0** to **99**)

The file name on the Symphony console is:

unitStats/<MUID><YYMMDDHHMMSS>.txt

4 Collected information

The following paragraphs list the various statistics that can be recorded by the WdpStats module.

4.1 Periodic

	Statistic name	Description	Value range
1	Battery	Battery charge	Integer, in percentage
2	WiFi_signal	WiFi signal level	Integer, in percentage
3	RAM	RAM available	Integer, en KB
4	StorageMemory	Storage memory available	Integer, en KB
5	FlashMemory	Flash memory available	Integer, en KB
6	CPU_Usage	CPU percentage usage	Integer, in percentage
7	Ping	Launch an ICMP Ping to a server via its IP address	Integer, response time in ms (-1 if not retrieved)
8	ActiveProcessQty	Quantity of active process in the PDA	Integer
9	ActiveThreadsQty	Quantity of active threads in the PDA	Integer
10	TestProcessActive	Test if some specific process is active, given its exe file	Integer, Quantity of active instance.
11	WiFi AccessPoint80211Type	Type of protocol 802.11 supported by the associated AP	1 : 802_11b 2 : 802_11a 3 : 802_11g
12	WiFi AccessPointChannel	Channel used by the associated Access Point	
13	WiFi AccessPointFrequency	Frequency used by the associated Access Point	
14	WiFi AssociationStatus	Association Status to some Access Point	0 : associated 1 : not associated 2 : roamed
15	WiFi ChannelQualityPercentMissedBeacons	Percentage of Missed Beacons in the last 5 seconds	Integer, in percentage
16	WiFi ChannelQualityPercentTxRetries	Percentage of retries of transmission in the last 2.5 seconds	Integer, in percentage
17	WiFi ChannelQualityRxRSSI	Last Rx RSSI	Integer
18	WiFi CountryCode	2 character country code	string
19	WiFi CurrentAccessPointMacAddress	MAC address of the associated AP	string
20	WiFi CurrentProfileName	Current Profile Name	string

Collected information

21	WiFi CurrentSSID	Current SSID	string
22	WiFi RetryCount	Retry Count in protocol 802.11	Integer
23	WiFi TransmitPower	Transmit power in μ W	Integer
24	WiFi TransmitRate	Transmit rate in Mbps	Integer
25	GPRS Signal Strength		Integer, in percentage
26	GPRS Equipment State		1: Minimum power state. 2: Only the receive radio is powered. 3: Only the transmit radio is powered. 4: Receive and transmit radio are off. 5: Full power.
27	GPRS Cnx Status	GPRS Connection Status	0 : not connected 1 : connected
28	GPRS bytes Tx	GPRS Quantity of bytes transmitted in the current connection.	Integer
29	GPRS bytes Rx	GPRS Quantity of bytes received in the current connection.	Integer
30	GPRS frames Tx	GPRS Quantity of frames transmit in the current connection.	Integer
31	GPRS frames Rx	GPRS Quantity of frames received in the current connection.	Integer
32	GPRS CRC errors	GPRS CRC errors in the current connection.	Integer
33	GPRS timeout errors	GPRS timeout errors in the current connection.	Integer
34	GPRS alignment errors	GPRS alignment errors in the current connection.	Integer
35	GPRS hardware overrun errors	GPRS hardware overrun errors in the current connection.	Integer
36	GPRS framing errors	GPRS framing errors in the current connection.	Integer
37	GPRS buffer overrun errors	GPRS buffer overrun errors in the current connection.	Integer
38	GPRS compression radio Rx		Integer
39	GPRS compression radio Tx		Integer
40	GPRS bps (cxn speed)		Integer
41	GPRS cnx time elapsed		Integer
42	GPS Position	Latitude, Longitude, Altitude and Speed	Integer

NOTE:

In the previous table, the following restrictions apply:

- The statistics numbers 1 to 10 are available for all terminal types.
- The statistics numbers 11 to 24 are available only for Symbol/Motorola with Fusion, Honeywell and PSION.
- The statistics numbers 25 to 41 are available for all terminal types with GPRS capability.

4.2 At event

	Event Name	Description	Value range
1	Reboot	Alert launched when the terminal warm/coldboots	(Date, Time, Type) Where Type means : in WinCE <ul style="list-style-type: none"> ▪ 0: warm boot ▪ 1: cold boot in Windows Mobile <ul style="list-style-type: none"> ▪ 0: warm/cold boot ▪ 1: clean boot
2	PowerOn	Alert launched when the terminal powers on	(Date, Time)
4	Network_Connect	Alert launched when the terminal connects to a network	(Date, Time)
5	Network_Disconnect	Alert launched when the terminal disconnects from a network	(Date, Time)
6	ACPower	Alert launched when the terminal connects to the AC power (charging)	(Date, Time)
7	Free falls	Event recorded when a free fall is detected	(Date, Time)

NOTE:

The statistic number 7 (Free falls) in the table above is available only on the following MOTOROLA devices: MC67NA, WT41N0 and VC70N0.

5 Transmission of the collected informations

All the statistics will be recorded in one or more files in the root folder with names:

```
\wdp.stats
or
\wdp0.stats
...
\wdp<n>.stats
```

The Symphony/WireLess Deployer Client will be in charge of sending this file every time it performs a LIST or an UPDATE command, repeated, scheduled or manual.

6 Schedule management

new Version 3.1.0.0

6.1 General considerations

Any statistic generated by WDPStats can have a **time schedule**. In other words it is possible to define time slices that define **when** the statistic is generated in the wdpXX.stats. This feature is particularly interesting to avoid meaningless statistics: for instance issuing a battery status when a device is on its docking station during the night. Moreover, a right configuration for the time schedules regarding the generated statistics will decrease the amount of data that is recorded and thus will improve the general performance of statistics queries and reports.

There are 10 time schedules available and each statistic can be set to follow one and only one time schedule among these 10. All the 10 time schedules are set in the [Main] section of the configuration file (wdpstats.cfg): the settings **ScheduleProfile00**, **ScheduleProfile01**,..., **ScheduleProfile09**.

Each time schedule is set on a week basis with possibly all the week days. Each day of the week can have up to 6 time slices. Here follows an example of a time schedule:

```
ScheduleProfile00=MON(08:30-18:00) TUE(08:30-18:00) WED(08:30-18:00) THU(08:30-18:00) FRI(08:30-18:00)
```

In order to link a statistic with a particular schedule, the **ScheduleIndex** or **ScheduleIndexXX** setting must be set to a value between 0 and 9 (both included) to reference one of the schedule among **ScheduleProfile00**, **ScheduleProfile01**,..., **ScheduleProfile09**.

6.2 Syntax

The following 6.2.x paragraphs describe the syntax for the Schedule management in the wdpstats.cfg configuration file. A full chapter is dedicated to complete examples ([6.3 Examples](#)).

6.2.1 Schedule profiles

All the possible schedules are stored in the **ScheduleProfile00**, **ScheduleProfile01**,..., **ScheduleProfile09** settings from the [Main] section of the configuration file wdpstats.cfg.

The syntax is the following (the non-terminal symbols are printed in bold and red, for example: **LUN**)

```
ScheduleProfileXX=DDD1(Time_slices1) DDD2(Time_slices2) ... DDDn(Time_slicesn)
```

where:

- **XX** is the current index inside a dynamic section (for example the [Battery] section) otherwise it is empty for “at event” statistics (the [Network_Connect] section for example),
- **DDD_i ::= LUN | MAR | MER | JEU | VEN | SAM | DIM | SUN | MON | TUE | WED | THU | FRI | SAT**
DDD_i is coding the day of the week with the following meaning:

Value	Day of the week
LUN	Monday
MAR	Tuesday
MER	Wednesday
JEU	Thursday
VEN	Friday
SAM	Saturday
DIM	Sunday
SUN	Sunday
MON	Monday
TUE	Tuesday
WED	Wednesday
THU	Thursday
FRI	Friday
SAT	Saturday

Each day of the week can be present at most onetime otherwise a syntax error is detected.

If a day of the week is not specified then the statistics that is linked with that schedule profile will not be issued from 00h00 till 23h59 for the relevant day.

- Each *Time_slices* element has the structure *time_slice(/time_slice)+* where

- *time_slice ::= [0-9][0-9]:[0-9][0-9]-[0-9][0-9]:[0-9][0-9]*

time_slice is coding a time slice. The first time is the time of the beginning of the slice and the second one (the one after the dash -) the end of the time slice.

Example: 08:30-18:25 is coding the time slice from 8h30 to 18h25.

There can be at most 6 *Time_slice* elements (separated by the pipe character |) for a given day.

Example: 08:30-12:30|13:00-17:00 is a *Time_slices* element that means the time slices from 8h30 till 12h30 and from 13h00 until 17h00.

The hours must be in the range [00, ..., 23] (bounds included) and the minutes in the range [00,...,59] (bounds included). If the of the two is not is the right range there will be a syntax error.

If a syntax error is detected on a schedule profile, any statistic linked to this profile will be issued from 00:00 to 23:59 for any day of the week which means the schedule profile is always active.

6.2.2 Schedule index

As specified above, a statistic is linked to a schedule profile by the index in 00,...,09.

For the the “At event” statistics, the setting **ScheduleIndex** set the index of the schedule profile.

For the the other statistics, the setting **ScheduleIndexXX** set the index of the schedule profile where *XX* is the index of the statistic itself.

A statistic can also be free of any schedule, which means it is issued at any time. In order to have a schedule profile free statistic, its **ScheduleIndex** or **ScheduleIndexXX** setting must be set to an invalid index value: -1.

6.3 Examples

In order to clearly illustrate how to configure the Schedule management, the present chapter shows several examples of configurations. For each of these examples the whole contents (ready for copy-and-paste) of the wdpstats.cfg file is printed and it is preceded by some explanations.

Example 1:

In the following example (complete wdpstats.cfg), only the GPS statistics is configured to follow a schedule. Indeed, the GPS statistic is linked with the schedule profile which has the index 0 (this profile is active every day from Monday to Friday, both included, from 08h30 to 18h00). All the others active/enabled statistics are not linked to any schedule (their schedule index is -1).

```
[Main]
MaxFileSize=20
MaxNumberOfFiles=10
OutputFileVersion=3
StatsDir=\
ExternalEventsVT=No
ExternalEventsS2=No
ExternalEventsFlex=No
ScheduleProfile0=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile1=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile2=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile3=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile4=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile5=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile6=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile7=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile8=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile9=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)

[Battery]
Counter=2
EventName00=Bat1
RetrieveInterval00=60
ThresholdType00=Both
Threshold00=0
Average00=5
ScheduleIndex00=-1
EventName01=BatThresMin
RetrieveInterval01=300
ThresholdType01=Min
Threshold01=20
Average01=1
ScheduleIndex01=-1

[MiFi_signal]
Counter=1
EventName00=MifiThMin
RetrieveInterval00=60
ThresholdType00=Min
Threshold00=20
Average00=1
ScheduleIndex00=-1

[RAM]
Counter=1
EventName00=RamThMin
RetrieveInterval00=60
ThresholdType00=Min
Threshold00=1024
Average00=1
ScheduleIndex00=-1

[StorageMemory]
Counter=1
EventName00=StoThMin
RetrieveInterval00=60
ThresholdType00=Min
Threshold00=2048
Average00=1
ScheduleIndex00=-1

[FlashMemory]
```

```
Counter=0

[CPU_Usage]
Counter=0

[Network_Connect]
EventName=Cnx
EventDo=Yes
ScheduleIndex=-1

[Network_Disconnect]
EventName=Disc
EventDo=Yes
ScheduleIndex=-1

[PowerOn]
EventName=PowOn
EventDo=Yes
ScheduleIndex=-1

[Reboot]
EventName=Reboot
EventDo=Yes
ScheduleIndex=-1

[ACPower]
EventName=AC
EventDo=Yes
ScheduleIndex=-1

[ Ping]
Counter=0

[ActiveProcessQty]
Counter=1
EventName00=Pro1
RetrieveInterval00=60
ThresholdType00=Both
Threshold00=0
Average00=1
ScheduleIndex00=-1

[ActiveThreadsQty]
Counter=0

[TestProcessActive]
Counter=0

[AccessPoint00211Type]
Counter=0

[AccessPointChannel]
Counter=0

[AccessPointFrequency]
Counter=0

[AssociationStatus]
Counter=0

[ChannelQualityPercentMissedBeacons]
Counter=0

[ChannelQualityPercentTxRetries]
Counter=0

[ChannelQualityRxRSSI]
Counter=0

[CountryCode]
Counter=0

[CurrentAccessPointMacAddress]
Counter=0

[CurrentProfileName]
Counter=0

[CurrentSSID]
Counter=0

[RetryCount]
Counter=0
```

```

[TransmitPower]
Counter=0

[TransmitRate]
Counter=0

[GPRS_SignalStrength]
Counter=0

[GPRS_EquipmentState]
Counter=0

[GPRS_ConnectionStatus]
Counter=0

[GPRS_CallInformation]
Counter=0

[GPSPosition]
EventName=GpsPos
EventDo=Yes
RetrieveInterval=60
ScheduleIndex=0

[Free_Fall]
EventName=Free_Fall
EventDo=No
ScheduleIndex=-1

```

Example 2:

In the following example (complete wdpstats.cfg), all the enabled statistics are NOT linked to ant schedule profile. Indeed, any active/enabled statistics has its schedule index set to -1.

```

[Main]
MaxFileSize=20
MaxNumberOfFiles=10
OutputFileVersion=3
StatsDir=\
ExternalEventsVT=No
ExternalEventsS2=No
ExternalEventsFlex=No
ScheduleProfile0=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile1=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile2=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile3=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile4=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile5=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile6=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile7=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile8=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile9=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)

[Battery]
Counter=2
EventName00=Bat1
RetrieveInterval00=60
ThresholdType00=Both
Threshold00=0
Average00=5
ScheduleIndex00=-1
EventName01=BatThresMin
RetrieveInterval01=300
ThresholdType01=Min
Threshold01=20
Average01=1
ScheduleIndex01=-1

[WiFi_signal]
Counter=1
EventName00=WifiThMin
RetrieveInterval00=60
ThresholdType00=Min
Threshold00=20
Average00=1

```

ScheduleIndex00=-1

[RAM]

Counter=1
EventName00=RamThMin
RetrieveInterval00=60
ThresholdType00=Min
Threshold00=1024
Average00=1
ScheduleIndex00=-1

[StorageMemory]

Counter=1
EventName00=StoThMin
RetrieveInterval00=60
ThresholdType00=Min
Threshold00=2048
Average00=1
ScheduleIndex00=-1

[FlashMemory]

Counter=0

[CPU_Usage]

Counter=0

[Network_Connect]

EventName=Cnx
EventDo=Yes
ScheduleIndex=-1

[Network_Disconnect]

EventName=Disc
EventDo=Yes
ScheduleIndex=-1

[PowerOn]

EventName=PowOn
EventDo=Yes
ScheduleIndex=-1

[Reboot]

EventName=Reboot
EventDo=Yes
ScheduleIndex=-1

[ACPower]

EventName=AC
EventDo=Yes
ScheduleIndex=-1

[Ping]

Counter=0

[ActiveProcessQty]

Counter=1
EventName00=Pro1
RetrieveInterval00=60
ThresholdType00=Both
Threshold00=0
Average00=1
ScheduleIndex00=-1

[ActiveThreadsQty]

Counter=0

[TestProcessActive]

Counter=0

[AccessPoint80211Type]

Counter=0

[AccessPointChannel]

Counter=0

[AccessPointFrequency]

Counter=0

[AssociationStatus]

Counter=0

[ChannelQualityPercentMissedBeacons]

Counter=0

[ChannelQualityPercentTxRetries]
Counter=0

[ChannelQualityRxRSSI]
Counter=0

[CountryCode]
Counter=0

[CurrentAccessPointMacAddress]
Counter=0

[CurrentProfileName]
Counter=0

[CurrentSSID]
Counter=0

[RetryCount]
Counter=0

[TransmitPower]
Counter=0

[TransmitRate]
Counter=0

[GPRS_SignalStrength]
Counter=0

[GPRS_EquipmentState]
Counter=0

[GPRS_ConnectionStatus]
Counter=0

[GPRS_CallInformation]
Counter=0

[GPSPosition]
EventName=GpsPos
EventDo=Yes
RetrieveInterval=60
ScheduleIndex=-1

[Free_Fall]
EventName=Free_Fall
EventDo=No
ScheduleIndex=-1

Example 3:

In the following example (complete `wdpstats.cfg`), there are more complex settings regarding the schedule management.

The 2 **battery statistics** (section `[Battery]`) have 2 different schedule profiles:

- The statistics named `Bat1` is active every day (from Monday till Friday) from 08:30 to 18:00.
- The statistics named `Bat2` is active whenever the `Bat1` statistic is NOT active which means between 18:00 to 08:30 the day after and the whole week-end.

The **Wifi signal strength** statistic (section `[Wifi_Signal]`, name `WifiThMin`) is active every day (from Monday till Friday) from 08:30 to 18:00.

The **RAM memory** statistic (section `[RAM]`, name `RamThMin`) and the **Storage memory** (section `[StorageMemory]`, name `StoThMin`) statistics are always active: every day of the week from 00:00 till 23:59.

The following statistics:

- **Network connection** (section `[Network_Connect]`, name `Cnx`)
- **Network disconnection** (section `[Network_Disconnect]`, name `Disc`)
- **Power On/resume from suspended state** (section `[PowerOn]`, name `PowOn`)
- **System reboot** (section `[Reboot]`, name `Reboot`)
- **AC power connected** (section `[ACPower]`, name `AC`)

are always active: every day of the week from 00:00 till 23:59

The following statistics:

- **Active Processes quantity** (section `[ActiveProcessQtty]`, name `Pro1`)
- **GPS coordinates** (section `[GPSPosition]`, name `GpsPos`)
- **Free falls detections** (section `[Free_Fall]`, name `Free_Fall`)

are active every day (from Monday till Friday) from 08:30 to 18:00.

```
[Main]
MaxFileSize=20
MaxNumberOfFiles=10
OutputFileVersion=3
StatsDir=\
ExternalEventsVT=No
```

```
ExternalEvents52=No
ExternalEventsFlex=No
ScheduleProfile00=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile01=LUN(00:00-08:30|18:00-23:59) MAR(00:00-08:30|18:00-23:59) MER(00:00-08:30|18:00-23:59) JEU(00:00-08:30|18:00-23:59) VEN(00:00-08:30|18:00-23:59) SAM(00:00-23:59) DIH(00:00-23:59)
ScheduleProfile02=LUN(00:00-23:59) MAR(00:00-23:59) MER(00:00-23:59) JEU(00:00-23:59) VEN(00:00-23:59) SAM(00:00-23:59) DIH(00:00-23:59)
ScheduleProfile03=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile04=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile05=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile06=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile07=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile08=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
ScheduleProfile09=LUN(08:30-18:00) MAR(08:30-18:00) MER(08:30-18:00) JEU(08:30-18:00) VEN(08:30-18:00)
```

[Battery]

```
Counter=2
EventName00=Bat1
RetrieveInterval00=60
ThresholdType00=Both
Threshold00=0
Average00=5
ScheduleIndex00=0
EventName01=BatThresMin
RetrieveInterval01=300
ThresholdType01=Min
Threshold01=20
Average01=1
ScheduleIndex01=1
```

[WiFi_signal]

```
Counter=1
EventName00=WifiThMin
RetrieveInterval00=60
ThresholdType00=Min
Threshold00=20
Average00=1
ScheduleIndex00=0
```

[RAM]

```
Counter=1
EventName00=RamThMin
RetrieveInterval00=60
ThresholdType00=Min
Threshold00=1024
Average00=1
ScheduleIndex00=2
```

[StorageMemory]

```
Counter=1
EventName00=StoThMin
RetrieveInterval00=60
ThresholdType00=Min
Threshold00=2048
Average00=1
```



```
ScheduleIndex00=2

[FlashMemory]
Counter=0

[CPU_Usage]
Counter=0

[Network_Connect]
EventName=Cnx
EventDo=Yes
ScheduleIndex=2

[Network_Disconnect]
EventName=Disc
EventDo=Yes
ScheduleIndex=2

[PowerOn]
EventName=PowOn
EventDo=Yes
ScheduleIndex=2

[Reboot]
EventName=Reboot
EventDo=Yes
ScheduleIndex=2

[ACPower]
EventName=AC
EventDo=Yes
ScheduleIndex=2

[ Ping]
Counter=0

[ActiveProcessQty]
Counter=1
EventName00=Pro1
RetrieveInterval00=60
ThresholdType00=Both
Threshold00=0
Average00=1
ScheduleIndex00=0

[ActiveThreadsQty]
Counter=0

[TestProcessActive]
Counter=0

[AccessPoint80211Type]
Counter=0
```

```
[AccessPointChannel]
Counter=0

[AccessPointFrequency]
Counter=0

[AssociationStatus]
Counter=0

[ChannelQualityPercentMissedBeacons]
Counter=0

[ChannelQualityPercentTxRetries]
Counter=0

[ChannelQualityRxRSSI]
Counter=0

[CountryCode]
Counter=0

[CurrentAccessPointMacAddress]
Counter=0

[CurrentProfileName]
Counter=0

[CurrentSSID]
Counter=0

[RetryCount]
Counter=0

[TransmitPower]
Counter=0

[TransmitRate]
Counter=0

[GPRS_SignalStrength]
Counter=0

[GPRS_EquipmentState]
Counter=0

[GPRS_ConnectionStatus]
Counter=0

[GPRS_CallInformation]
Counter=0

[GPSPosition]
```

```
EventName=GpsPos  
EventDo=Yes  
RetrieveInterval=60  
ScheduleIndex=0
```

```
[Free_Fall]  
EventName=Free_Fall  
EventDo=No  
ScheduleIndex=0
```

7 Configuration

The configuration is set by a file on the device: WdpStats.cfg.

This configuration file is a Microsoft .INI file where each section sets a statistic.

7.1 Periodical statistics with or without threshold

Each of them has the following keys:

- **RetrieveInterval**
- **Threshold**
- **ThresholdType**
- **Average**
- **EventName**

In general **Threshold = 0** means that **Threshold** is not used. **RetrieveInterval** is the interval to use for retrieving the data expressed in seconds.

AverageType can be **Min** (data passed the threshold from high to low), **Max** (data passed the threshold from low to high) or **Both**.

If **Average** is 1, data is registered every time it is retrieved. An **Average** of n greater than 1 means that data is retrieved n times: Value_1, Value_2, ... Value_n and the value registered is an average of them [(Value_1+Value_2+...+Value_n) / n].

There are some sections where the quantity of keys is more or less than 4:

- **Ping**: with the common 4 keys and **IPHost**, the IP address of the host.
- **TestProcessActive**: with the common 4 keys and **ExeFile**, the name of the file containing the executable of the launched process.
- **CountryCode** gives results as a string, with only **RetrieveInterval** .
- **CurrentAccessPointMacAddress** gives results as a string, with only **RetrieveInterval** .
- **CurrentProfileName** gives results as a string, with only **RetrieveInterval** .
- **CurrentSSID** gives results as a string, with only **RetrieveInterval** .

7.2 Periodical statistics with or without change

Each of them has the following keys:

- **RetrieveInterval**
- **OnChange**
- **EventName**

These events can be retrieved on interval and report the result at each retrieve, or only when the value changes of state.

7.3 At Event

Each of them has the following keys:

- **EventDo**
- **EventName**

These events have no interval and are reported when the event occurs (i/e reboot).

7.4 Main Section

The Main Section has some parameters.

- **MaxFileSize** (maximum of the size of local files wdp[N].stats in KB)
- **MaxNumberOfFiles** (maximum for the quantity of files wdp<N>.stats).

In such a way, the quantity of KB used in the PDA will be **MaxFileSize*MaxNumberOfFiles**. The **MaxNumberOfFiles** files will be rewritten in a circular way. The old files will be erased if needed.

- **StatsDir** is the directory where the stats file will be stored
- **OutputFileVersion** is the version of the output file (for compatibility with WireLess Deployer)

7.5 Configuration file sample:

```
%% Wdp Stats
%% Version 2.0
%% Configuration File

[Main]
MaxFileSize = 10
MaxNumberOfFiles =5

%% Test if the battery level passed from a value greater
%% than 40% to a value less than 40%
%% Test it every 5 minutes
[Battery]
Counter = 3
RetrieveInterval00 = 300
ThresholdType00 = Min
Threshold00 = 40
Average00 = 1
EventName00      = Bat1

%% send an alert immediately
%% when the battery level is less than 20%
%% test its value every 5 minutes
RetrieveInterval01 = 300
ThresholdType01 = Min
Threshold01 = 20
Average01 = 1
EventName00      = Bat2

%% Test the battery level every 5 minutes
%% Make hourly averages
RetrieveInterval02 = 300
ThresholdType02 = Min %% (no sense here, Threshold is 0)
Threshold02 = 0
Average02 = 12
EventName00      = Bat3

%% retrieve RAM usage every 5 minutes,
[RAM]
Counter = 1
RetrieveInterval00 = 300      %% in seconds, 5 min
ThresholdType00 = Min
Threshold00 = 0
Average00 = 1
EventName00      = Ram1

%% retrieve WiFi signal every 5 minutes
%% register every hour an average of 12 values
[WiFi_signal]
Counter = 1
RetrieveInterval00 = 300      %% in seconds, 5 min
ThresholdType00 = Min
Threshold00 = 0
```

```

Average00 = 12
EventName00      = Wif1

%% send information about reboots
%% RetrieveInterval, Threshold etc mean nothing here
[Reboot]
EventDo          = Yes
EventName        = Boo

%% ... it continues
%% perform a ping to a host every 5 minutes,
[Ping]
Counter = 2
RetrieveInterval00 = 300          %% in seconds, 5 min a
                                %%                [ping is performed]

ThresholdType00 = Min
Threshold00 = 0
Average00 = 1
IPHost00 = 10.10.10.19
EventName00      = Ping1

%% perform a ping to a host every 5 minutes,
%% register to the file only if the host is unreachable
IPHost01 = 192.168.1.4
RetrieveInterval01 = 300          %% in seconds, every 5 min a
                                %%                [ping is performed]

ThresholdType01 = Min
Threshold01 = -1
Average01 = 1
EventName01      = Ping2

%% retrieve a CPU usage every 5 minutes,
%% register to the file only if it is more than 70%
[CPU_Usage]
Counter = 1
RetrieveInterval00 = 300          %% in seconds, every 5 min a
                                %%                [test is performed]

ThresholdType00 = Max
Threshold00 = 70
Average00 = 1
EventName00      = Cpu1

%% do not retrieve active threads information.
[ActiveThreadsQtty]
Counter = 1
RetrieveInterval00 = 0
ThresholdType00 = Min
Threshold00 = 0
Average00 = 1
EventName00      = Thr1

%% ... etc.

```

In this example, it is possible to see the different event types (PowerOn)

A statistic line is generated in the output file every time the threshold is reached or overstepped, but not when it continues having the state below (or above) the bound.

The file wfb[N].stats, when sent to the server, becomes MUIDYYYYMMDDhhmmss.stats, and will be read by the process in charge of filling the DB for reporting.

8 wdp.stats Structure

When the value of **Average** is different from 1 in the configuration file, the generated file contains (besides **Result**) a **Quantity** value: the number of values retrieved until overpassing the threshold **Average** times.

The first line is always one and only one of the 3:

- **WDP STATS VERSION 1.1**
- **WDP STATS VERSION 2.0**
- **WDP STATS VERSION 3.0**

according to the output file version.

8.1 V1.1 Lines

Field	Type	Size	Description
Timestamp	Timestamp	14	YYYYMMDDhhmmss
RetrieveInterval	int	var	From configuration, but rounded to minutes
RegType	By now: B for Battery Level W for RF signal level	1	Record id
Threshold	int	var	From configuration 0 to 100
ThresholdType	M or m or B	1	From configuration: M for Max , m for Min , B otherwise
Average	int	var	From configuration.
Separator	Separator	1	Always ';' (semicolon)
Quantity (only reported if Threshold is not 0)	int	var	Quantity of values retrieved before calculating Result (it may defer from Average if Threshold is not 0)
Separator (only if Threshold is not 0)	Separator	1	Always ';' (semicolon)
Result	int	var	

8.2 V 2.0 lines

Field	Type	Size	Description
Timestamp	Timestamp	14	YYYYMMDDhhmmss
Retrieve Interval	Int	var	From the configuration, but rounded to minutes
Event Type	(See event list)	2	
Threshold	int	var	From config. 0 à 100
ThresholdType	+ or - or =	1	From config: + for Max , - for Min , = in other case
Average	Int	var	From config.
Separator	Separator	1	Always ';' (semicolon)
Quantity (only reported if Threshold is not 0)	int	var	Quantity of values retrieves before calculating "Result" (it would be different of Average if Threshold is not 0)
Separator (only if Threshold is not 0)	Separator	1	Always ';' (semicolon)
Result	int	var	One or the other depending on RegType
strResult	string	var	

8.3 V 3.0 Header

Field	Type	Size	Description
Event Number	int	var	1-n (event id)
Separator	Separator	1	Always ';' (semicolon)
Type	String	2	Event type and subtype (see event list)
Separator	Separator	1	Always ';' (semicolon)
Name	String	var	Event configured name
Separator	Separator	1	Always ';' (semicolon)
Retrieve interval	Int	var	Retrieve interval, rounded to minutes
Separator	Separator	1	Always ';' (semicolon)
Threshold Type	Char	1	From configuration: + for Max , - for Min, * max always / min always = both
Separator	Separator	;	Always ';' (semicolon)
Threshold	int	var	Configured threshold type
Separator	Separator	;	Always ';' (semicolon)
Average	int	var	From configuration.
Line separator	Separator	2	Always 0x0d 0x0a (CR LF)

8.4 V3.0 Data header

Field	Type	Size	Description
Data Indicator	string	3	"..."
Line separator	Separator	2	Always 0x0d 0x0a (CR LF)

8.5 V3.0 Data line

Field	Type	Size	Description
Timestamp	Timestamp	14	YYYYMMDDhhmmss
Separator	Separator	1	Always ';' (semicolon)
Event Number	int	var	1-n (event id)
Separator	Separator	1	Always ';' (semicolon)
Value	int or String	var	Event value
Line separator	Separator	2	Always 0x0d 0x0a (CR LF)

8.5.1 GPRS statistics

The following paragraphs add precisions about the GPRS statistics except for the "g0" statistics (GPRS signal strength) which is a numeric value.

8.5.1.1 g1

The following table describes the possible values for the "g1" statistics for GPRS equipment state.

Value	Description
1	Minimum power state.
2	Only the receive radio is powered.
3	Only the transmit radio is powered.
4	Receive and transmit radio are off.
5	Full power.

8.5.1.2 g2

The current section specifies the format of the Value for the "g2" GPRS statistic (GPRS connection status) in the table below:

Value	Description
0	The GPRS connection status is NOT connected.
Different from 0	The GPRS connection status is CONNECTED.

8.5.1.3 g3

The current section specifies the format and syntax of the "g3" GPRS statistic data/value. The Value part of this statistic is made of 14 numeric values separated by a semi-colon (';').

The following table is the list of these values. The index column indicates their order inside the Value field:

Index	Meaning
1	The number of bytes transmitted through this connection or link.
2	The number of bytes received through this connection or link.
3	The number of frames transmitted through this connection or link.
4	The number of frames received through this connection or link.
5	The number of cyclic redundancy checks (CRC) errors on this connection or link.
6	The number of timeout errors on this connection or link.
7	The number of alignment errors on this connection or link.
8	The number of hardware overruns errors on this connection or link.
9	The number of framing errors on this connection or link.
10	The number of buffer overruns errors on this connection or link.
11	The compression ratio for the data being received on this connection or link.
12	The compression ratio for the data being transmitted on this connection or link.
13	The speed of the connection or link, in bits per second. This speed is negotiated at the time that the connection or link is established.
14	The amount of time, in milliseconds, that the connection or link has been connected.

9 Examples

Given this configuration file:

```

%% Test the battery level every 5 minutes
%% Make hourly averages
[Battery]
Counter = 1
RetrieveInterval00 = 300
ThresholdType00 = Min          %% redundant, no sense
Threshold00 = 0
Average00 = 12
EventName00 = Bat1

%% retrieve WiFi signal every 5 minutes
%% register an alert if signal passed through 60%
[WiFi_signal]
Counter = 1
RetrieveInterval00 = 300
ThresholdType00 = Both
Threshold00 = 60
Average00 = 12
EventName00 = Wi1

```

the output statistics file could contain:

```

WDP STATS VERSION 1.1
200810250910255B0m12;78
200810251010465B0m12;74
200810251110585B0m12;69
200810251311255B0m12;49
200810251411375B0m12;41
200810251512125W60B10;48;58
200810251511485B0m12;38

```

or

```

WDP STATS VERSION 2.0
200810250910255 B0-12;78
200810251010465 B0-12;74
200810251110585 B0-12;69
200810251311255 B0-12;49
200810251411375 B0-12;41
200810251512125 W60=10;48;58
200810251511485 B0-12;38

```

or

```

SPN STATS VERSION 3.0
1; B;Bat1;5;-;0;12
2; W;Wi1;5;=;60;12
...
20120315151148;1;62
20120315161204;1;60
20120315161304;2;38
20120315161304;1;47

```

10 List of statistics IDs for V3 output

In the following tables, all the statistics are summarized:

- The first column labeled "Event ID" is a 2 characters string enclosed by double quotes. The first character codes the event type and the second character codes the event sub-type.
- The second column labeled "Meaning" gives a short description the statistics
- The third section labeled "Section" is the name of the corresponding section in the configuration file (wdpstats.cfg).



Warning

Please remind that the following tables (§9.1 to 9.6) are only relevant for the **V3** version of the output statistics file.

10.1 General statistics

Event ID	Meaning	Section
" B"	Battery Level	Battery
" W"	RF signal level (WiFi)	WiFi_Signal
" A"	AC Power	ACPower
" C"	Connection	Network Connect
" D"	Disconnection	Network Disconnect
" E"	Test if a particular process is active	TestProcessActive
" F"	Flash Memory	FlashMemory
" G"	Ping	Ping
" I"	Reboot	Reboot
" K"	Active processes quantity	ActiveProcessQty
" L"	Active threads quantity	ActiveThreadsQty
" P"	Power On	PowerOn
" R"	RAM	RAM
" S"	Storage Memory	StorageMemory
" U"	CPU Usage	CPU_Usage

10.2 Wi-Fi statistics

Event ID	Meaning	Section
"X0"	802.11 access point type	AccessPoint80211Type
"X1"	Wi-Fi access point channel	AccessPointChannel
"X2"	Wi-Fi access point frequency	AccessPointFrequency
"X3"	Wi-Fi association status	AssociationStatus
"X4"	Wi-Fi channel quality: percent of missed beacons	ChannelQualityPercentMissedBeacons
"X5"	Wi-Fi channel quality: percent of Tx retries	ChannelQualityPercentTxRetries
"X6"	Wi-Fi channel quality: RxRSSI	ChannelQualityRxRSSI
"X7"	Wi-Fi country code	CountryCode
"X8"	Wi-Fi current access point MAC address	CurrentAccessPointMacAddress
"X9"	Wi-Fi current profile name	CurrentProfileName
"XA"	Wi-Fi current SSID	CurrentSSID
"XB"	Wi-Fi retry count	RetryCount
"XC"	Wi-Fi transmit power	TransmitPower
"XD"	WiFi transmit rate	TransmitRate

10.3 WireLess Telnet statistics

Event ID	Meaning	Section
"FC"	TelNet 5250 Connection event	Main (see the note below)
"FD"	TelNet 5250 Disconnection event	Main (see the note below)
"FS"	TelNet 5250 Barcode scans quantity	Main (see the note below)
"FK"	TelNet 5250 Keys quantity	Main (see the note below)
"FA"	TelNet 5250 Average response time in ms	Main (see the note below)
"FF"	TelNet 5250 Total frames	Main (see the note below)
"FT"	TelNet 5250 Timeout	Main (see the note below)
"TC"	TelNet VT Connection event	Main (see the note below)
"TD"	TelNet VT Disconnection event	Main (see the note below)
"TS"	TelNet VT Barcode scans quantity	Main (see the note below)
"TK"	TelNet VT Keys quantity	Main (see the note below)
"TT"	TelNet VT timeout	Main (see the note below)
"TA"	TelNet VT Average response time in ms	Main (see the note below)
"TF"	TelNet VT Total frames	Main (see the note below)



Note

The WireLess Telnet events in the above table are not affected by the time schedule. It is not possible to bind one of the WireLess Telnet to a time schedule profile.

10.4 GPRS statistics

Event ID	Meaning	Section
"g0"	GPRS Signal Strength	GPRS_SignalStrength
"g1"	GPRS Equipment State	GPRS_EquipmentState
"g2"	GPRS Connection Status	GPRS_ConnectionStatus
"g3"	GPRS Call informations	GPRS_CallInformation

10.5 GPS statistics

Event ID	Meaning	Section
"p0"	GPS position	GPSPosition

10.6 Hits and falls statistics

Event ID	Meaning	Section
"MF"	Free fall event	Free_Fall

11 Platform versions of WdpStats V3

WdpStats is compiled for different hardware platforms. Depending on these platforms, some statistics may be available or not. The following paragraphs list the statistics whose availability may change according to the hardware platform.

11.1 Wifi

	Event Name	Generic	Motorola Fusion	Intermec	Honeywell	Summit	PSION
2	W WiFi_signal	✓	✓	✓	✓	✓	✓
11	X0 WiFi AccessPoint80211Type		✓				✓
12	X1 WiFi AccessPointChannel		✓		✓	✓	
13	X2 WiFi AccessPointFrequency		✓				✓
14	X3 WiFi AssociationStatus		✓	✓	✓	✓	✓
15	X4 WiFi ChannelQualityPercentMissedBeacons		✓				
16	X5 WiFi ChannelQualityPercentTxRetries		✓				
17	X6 WiFi ChannelQualityRxRSSI		✓				
18	X7 WiFi CountryCode		✓				
19	X8 WiFi CurrentAccessPointMacAddress	✓	✓		✓	✓	✓
20	X9 WiFi CurrentProfileName		✓			✓	
21	XA WiFi CurrentSSID		✓		✓	✓	✓
22	XB WiFi RetryCount		✓				
23	XC WiFi TransmitPower		✓			✓	
24	XD WiFi TransmitRate		✓	✓	✓	✓	

11.2 Free fall

The "Free fall" statistic is available only on the following MOTOROLA devices:

- MC67NA
- MC32N0,
- WT41N0
- VC70N0.

As a result, the "Free fall" statistic is only working on the **Motorola Fusion** hardware platform.