Porting Guide From eDSoft V3.10 to WIPSoft V2.01

Revision: 003
Date: January 12, 2007

OPEN AT

wavecom
Make it wireless

Operating Systems | Integrated Development Environments | Plug-Ins | Wireless CPUs | Services

Porting Guide From eDSoft V3.10 to WIPSoft V2.01

Reference:	WM_DEV_OAT_UGD_027
Revision:	003
Date:	January 12, 2007

Trademarks

®,WAVECOM®, WISMO®, Open AT® and certain other trademarks and logos appearing on this document, are filed or registered trademarks of Wavecom S.A. in France or in other countries. All other company and/or product names mentioned may be filed or registered trademarks of their respective owners.

Copyright

No patent liability is assumed with respect to the use of the information contained herein.

Overview

The aim of this document is to provide Wavecom customers with a full description of the APIs associated with the Open AT® IP Connectivity library.

Document History

Level	Date	History Of the Evolution	Writer
001	May 22 2006	Creation	Wavecom
002	August 7 2006	Preliminary	Wavecom
003	January 12 2007	Final	Wavecom

Contents

1 Introduction

1.1 Related Documents

[1] WIP Open AT IP Connectivity Development Guide (ref WM_DEV_OAT_UGD_021).

[2] AT Commands Interface Guide for IP Connectivity (eDSoft V3.10) (ref WM_ASW_OAT_UGD_011 revision 007).

[3] WIP Open AT Commands User Guide (WM_DEV_OAT_UGD_024)

1.2 Abbreviations

APN	Access Point Name
CMUX	Converter Multiplexer
DLE	Escape character having a hex value 0x10
DNS	Domain Name Server
ETX	Escape character having a hex value 0x03.
FTP	File Transfer Protocol
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
IP	Internet protocol
ISP	Intenret Service Provider
OSI	Open System Interconnection
POP	Post Office Protocol
PPP	Point to Point Protocol
PSTN	Public Switched Telephone Network
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver Transmitter

1.3 Glossary

AT Mode	The functioning mode of UART, in which anything that is received from UART is treated as AT command.
Bearer	The term used in WIPSoft. It is used to indicate the layer providing the actual transmission of data from one peer to another
Data Mode	The functioning mode of UART, in which everything that is received from the UART is treated as data
eDSoft	An Open AT® application (providing the IP connectivity function to the Wireless CPU®) written using the eDLib IP library
UART	Universal Asynchronous Receiver Transmitter
WIPSoft	An Open AT® application (providing the IP connectivity function to the Wireless CPU®) written using the Wavecom IP library
+++	The escape sequence surrounded with 1 second delay which is used to switch the state of UART from data mode to AT mode

2 AT Commands

This chapter lists all the commands that have been introduced with the WIPSoft along with their usage and functionality. This chapter provides comparison between the commands available in eDSoft and WIPSoft.

2.1 Introduction

The WIPSoft application provides a more consistent interface to the user. The WIPSoft uses the APIs provided by WIPLib and provides custom AT command interface to the external application. This is an Open AT® application that implements the TCP/IP protocols using custom AT commands. This Open AT® application operates in co-operative mode and must be downloaded to the Wavecom Wireless CPU®. The commands are sent from an external application and the corresponding responses are sent back from the Wavecom Wireless CPU® to the external application.

The WIPSoft application maintains a set of protocol identifiers for supported protocols. These identifiers along with the protocol name are listed below in the table:

Protocol Identifier	Protocol
1	UDP protocol
2	TCP socket in client mode
3	TCP socket in server mode
4	FTP protocol

2.2 TCP/IP Services

2.2.1 TCP/IP Services in eDSoft

The eDSoft application supports multiple services to run at the same time. However, only one instance of a particular service could run at a time. The only exception to this rule is TCP socket service, where 2 sockets could be open at a time. Multiplexing of various services is done using the commands which are used to manipulate the service being used.

2.2.2 TCP/IP Services in WIPSoft

The WIPSoft allows concurrent execution of many services like TCP, UDP and FTP. However, the number of sockets for TCP and UDP and the number of sessions for FTP are limited. At a time, WIPSoft supports the following:

Protocol	Number of Sockets/Sessions
`UDP`	8
`TCP client`	8
`TCP server`	4
`FTP session`	1

Multiplexing between various services is achieved using the commands which are used to manipulate the service.

2.3 Error Codes

2.3.1 Error Codes in eDSoft

The eDSoft application provides the errors in the following format:

```
#CME ERROR: <Error Code>
```

The Error Code can take values from 34817 to 49158. The following table depicts the description of various error codes.

Error Code	Description of Error code
34817	Bad command: Unknown command
34819	Bad command: Syntax error
34824	Bad command: Write failed
34881	Bad command: Command too long
34882	Bad command: Bad command argument value
34883	Bad command: High level internet configuration only command
35840	Physical layer: Modem is already running
35841	Physical layer: GPRS connection lost
35862	Physical layer: Timeout, no activity on network connection
35865	Physical layer: Module is not attached to the network
35866	Physical layer: Invalid event during activation process
35867	Physical layer: Physical layer connection is currently not active
35868	Physical layer: GPRS connection aborted
35869	Physical layer: Invalid incoming call type
35870	Physical layer: Incoming call CLI not provided
35871	IP Connectivity library: SIM removed
36872	IP Connectivity library internal error: internal resource unavailable
36929	IP Connectivity library: Bad parameter configuration attempt
37120	IP Connectivity library: PPP negotiation failed (client configuration)
37121	IP Connectivity library: PPP negotiation failed (server configuration)
37122	IP Connectivity library: Another internal application is already running

Error Code	Description of Error code
37123	IP Connectivity library: Service is running. Unable to set parameter
37952	Distant: TCP session closed (TCP Context cancelled)
37964	Distant: No response from server
37966	Distant: TCP session closed by peer (FIN received from peer)
38016	Distant: Open session attempt failed
38017	Distant: Data send attempt failed
38018	Distant: Close session attempt failed
38022	Distant: Change Directory attempt failed
38023	Distant: File deletion attempt failed
38024	Distant: Data retrieve attempt failed
38025	Distant: Email retrieve attempt failed
38026	Distant: Email header receive failed
38027	Distant: No answer from DNS servers or the domain name resolution could not be completed by the server
38028	Distant: Sender email address rejected by server
38029	Distant: Recipient email address rejected by server
38030	Distant: CC Recipient email address rejected by server
38031	Distant: Email body send request rejected by server
38080	Distant: Username rejected by server
38081	Distant: Password rejected by server
38980	IP Connectivity library: PPP timeout (client configuration)
38981	IP Connectivity library: PPP timeout (server configuration)
49153	Internal error: Open data flow request failed
49154	Internal error: Close data flow request failed
49155	Internal error: Open GPRS session request failed
49156	Internal error: GPRS authentication failed
49157	Internal error: GPRS get IPCP information request failed
49158	Internal error: Open flow confirmation not received

2.3.2 Error Codes in WIPSoft

The WIPSoft application provides the error codes in the standard AT response format. Hence, if +CME error code generation is not enabled, a simple "ERROR" message is returned. In case, the +CME ERROR messages are enabled using +CMEE=1 command, the error codes takes the following format:

```
+CME ERROR: <Error Code>
```

The error code can have values from 800 to 818. The following table depicts the description of various error codes.

Error Code	Description of Error code
800	Invalid option specified
801	Invalid option value
802	Not enough memory
803	Operation not allowed in current stack state
804	Device already open
805	Network interface not available
806	Operation not allowed on the selected bearer
807	Bearer connection failure: line busy
808	Bearer connection failure: no answer
809	Bearer connection failure: no carrier
810	Bearer connection failure: no SIM card present
811	Bearer connection failure: SIM not ready (SIM PIN not given)
812	Bearer connection failure: GPRS network failure
813	Bearer connection failure: PPP LCP negotiation failed
814	Bearer connection failure: PPP authentication failed
815	Bearer connection failure: PPP IPCP negotiation failed
816	Bearer connection failure: PPP peer has terminated the session
817	Bearer connection failure: PPP peer not answering to echo requests
818	Incoming call refused
819	Error on Ping channel
820	Error writing configuration in FLASH memory
821	Error reading configuration in FLASH memory

Error Code	Description of Error code
822-829	Reserved for future use
830	Bad index
831	Bad state
832	Bad port number
833	Bad port state
834	Not implemented
835	Option not supported
836	Memory
837	Bad protocol
838	No more free socket
839	Error during channel creation
840	FTP session already active
841	Peer closed
842	Destination host unreachable (whether host unreachable, Network unreachable, response timeout)
843-849	Reserved for future use
850	Unknown reason
851	Bad state

2.4 Configuration Commands

The WIPSoft application allows better control of IP stack. This is achieved using a set of configurable options available with various WIP AT commands.

2.4.1 IP Stack Initialization and Termination (New)

2.4.1.1 Description

The WIPSoft application provides +WIPCFG command for IP stack initialization and termination. The eDSoft application does not support any initialization command. In eDSoft, the IP stack is initialized when application starts.

2.4.1.2 Syntax

```
AT+WIPCFG=<Option>
```

2.4.1.3 Defined Values

Option:

<Option>	Description
1	Start IP stack
0	Stop IP stack

2.4.1.4 Examples

Commands	Possible responses
AT+WIPCFG=1 *Note: Start the IP stack*	OK
AT+WIPCFG=0 *Note: Stop the IP stack*	+CME ERROR: 802 *Note: Stop procedure failed.*

2.4.2 IP Stack Configuration (New)

The WIPSoft application provides command to configure the values for internal parameters being used by the IP stack. These parameters include

- TTL (Time To Live of IP datagram)
- TOS (Type of Service)
- IP fragment timeout and so on

2.4.2.1 Description

The WIPSoft application provides +WIPCFG command for configuring the internal parameters of IP stack.

2.4.2.2 Syntax

```
AT+WIPCFG=2,<opt num>,<value>
```

Refer to [3] for more information about the parameters and defined values and the examples.

2.4.3 Retrieving IP Stack Configuration (New)

2.4.3.1 Description

The WIPSoft application provides AT+WIPCFG? command for retrieving the internal parameters of IP stack.

2.4.3.2 Syntax

```
AT+WIPCFG?
```

Refer to [3] for more information on the examples.

2.4.4 IP Stack Configuration Management (New)

2.4.4.1 Description

The WIPSoft application provides +WIPCFG command for storing configuring parameters in FLASH memory.

2.4.4.2 Syntax

```
AT+WIPCFG=4,<mode>
```

Refer to [3] for more information about the parameters and defined values and the examples.

2.5 Bearer Management

2.5.1 Additional Bearers (New)

The WIPSoft introduces a concept of a generic “Bearer”. A “Bearer” actually means a layer which would bear/receive the data sent to it by the IP layer and would forward it to the network. The “Bearer” can be correlated with the physical layer that is present in the OSI layer model.

In eDSoft, only two bearers are available to establish socket connections. These bearers are:

- GSM bearer: This bearer indicates that a GSM data call will be used to establish the IP connectivity. In this case, GSM data call will act as the physical layer.
- GPRS bearer: This bearer indicates that GPRS session will be used to establish the IP connectivity. In this case, GPRS session will act as the physical layer.

The WIPSoft application extends the above mentioned scenario and provides more bearers using which the IP layer connectivity can be established. The bearers that are available in WIPSoft are:

- GSM bearer: The GSM data call (as mentioned above)
- GPRS bearer: The GPRS bearer (as mentioned above)
- UART1: UART1 can also be used to establish an IP layer connection. A external device (For e.g. PC) can be connected to the Wireless CPU® to transfer TCP/IP data
- UART2: UART2 is used to establish the IP layer connection. This indicates that the client/server is running on the external microprocessor connected to UART2.
- CMUX ports over UARTs: The CMUX ports can also be used to establish the IP layer connection.

2.5.1.1 Associated Commands +WIPBR

2.5.1.1.1 Description

The WIPSoft application provides +WIPBR command to select and open a new available bearer such as UART.

2.5.1.1.2 Syntax

```
AT+WIPBR=1,<bid>
```

Refer to [3] for more information about the parameters and defined values and the examples.

2.5.2 Bearer Configuration Commands

2.5.2.1 Deprecated Commands

The following configuration commands are not available in WIPSoft:

- AT#ANSWERMODE
- AT#CALLBACKTIMER
- AT#CALLSCREENNUM
- AT#PHYTIMEOUT
- AT#DIALN2
- AT#DIALSELECT
- AT#REDIALCOUNT
- AT#REDIALDELAY
- AT#PPPPEERIP=<IP>

2.5.2.2 Selecting the GSM/GPRS Bearer

The +WIPBR allows to select between GSM and GPRS bearer.

Old interface
AT#GPRSMODE=<mode> *//Select GSM/GPRS bearer*
New interface
AT+WIPBR=1,<bid> *// Select GSM/GPRS bearer with additional parameters*

Refer to [3] for more information about the parameters and defined values and the examples.

2.5.2.3 Configuring the PPP Mode

The +WIPBR command can be used to configure the PPP mode (client/server)

Old interface
AT#PPPMODE=<mode> *//Select client/server*

New interface
AT+WIPBR=4,<bid>,<mode>,<other params> *// Mode determines the client or server*

The <other params> field can take the values depending on the <mode> and the bearer type as defined in the following table.

Bid	Mode	Other params
1..3,11..14,21..24	0	<none>
1..3,11..14,21..24	1	<PPP login>, <PPP password>
5	0	<none>
5	1	<login>,<password>[,<caller identity>]
6	0	<none>

NOTE	Several bearer can be opened at the same time but only one bearer can be started at a time

2.5.2.3.1 Configuring ISP Parameters

The +WIPBR command allows to configure the

- number to dial
- user name
- password

Old interface
AT#DIALN1=<number to dial> **AT#ISPUN=<user name>** **AT#ISPPW=<password>**
New interface
AT+WIPBR=2,5,2,<number to dial> **AT+WIPBR=2,5,0,<user name>** **AT+WIPBR=2,5,1,<password>**

2.5.2.4 Configuring the GSM PPP Server Bearer

2.5.2.4.1 Configuring Ring Counts

The +WIPBR command allows to configure the ring count. The WIPSoft command does not allow automatic accept but provides the ring indication only after ring counts specified in <value> parameter have elapsed.

Old interface
AT#RINGCOUNT=<value>
New interface
AT+WIPBR=2,5,5,<value>

2.5.2.4.2 Configuring IP Address of PPP Server

The +WIPBR command allows to configure the IP address assigned to Wireless CPU® itself when in PPP server mode.

Old interface
AT#PPPMYIP=<IP>
New interface
AT+WIPBR=2,<bid>,15,<IP>

2.5.2.4.3 Configuring ISP Authentication Parameters

The +WIPBR command allows to configure the username and password for PPP server. This authentication details should be used by PPP client while connecting to PPP server.

Old interface
AT#PPPSERVUN=<username> **AT#PPPSERVPW=<password>**
New interface
AT+WIPBR=2,5,0,<username> **AT+WIPBR=2,5,1,<password>**

2.5.2.5 Configuring the GPRS Bearer Parameters

The +WIPBR command allows to configure the access point related parameters for GPRS. These parameters include:

- access Point name
- user name
- password
- context id

Old interface
`AT#APNSERV=<APN>`
`AT#APNUN=<username>`
`AT#APNPW=<password>`
`AT#GPRSCID=<Context id>`
New interface
`AT+WIPBR=2,6,11,<APN>`
`AT+WIPBR=2,6,12,<Context id>`
`AT+WIPBR=2,6,0,<username>`
`AT+WIPBR=2,6,1,<password>`

2.5.3 Connection Management Commands

2.5.3.1 Deprecated Commands

The following configuration commands are not available in WIPSoft:

- AT#ACCEPT

2.5.3.2 Start the Bearer

The +WIPBR can be used to start the TCP/IP connection procedure.

Old interface
AT#CONNECTIONSTART
New interface
AT+WIPBR=4,<bid>,<mode>,<other params> *//parameter value "4" is used to start the connection procedure*

2.5.3.3 Stop the Bearer

The +WIPBR can be used to stop the active or outgoing connection.

Old interface
AT#CONNECTIONSTOP
New interface
AT+WIPBR=5,<bid> *//parameter value "5" is used to stop the connection procedure*

2.5.4 Miscellaneous Commands

2.5.4.1 Displaying IP Address

The +WIPBR command can be used to get the current IP address

Old interface
AT#DISPLAYIP
New interface
AT+WIPBR=3,<bid>,15 *//parameter value "15" is used to get the local IP address*

2.5.4.2 Displaying PPP Parameters

The +WIPBR command can be used to get the current PPP parameters

Old interface
AT#VPPP
New interface
AT+WIPBR=3,5,0 *//Username* **AT+WIPBR=3,5,1** *//Password* **AT+WIPBR=3,5,5** *//Ring count* **AT+WIPBR=3,5,15** *//Local IP address* **AT+WIPBR=3,5,16** *//Peer IP address*

2.5.4.3 Displaying GPRS Parameters

The +WIPBR command can be used to get the current GPRS parameters

Old interface
AT#VGPRS
New interface
AT+WIPBR=3,6,0 *//Username* **AT+WIPBR=3,6,1** *//Password* **AT+WIPBR=3,6,12** *//Context id* **AT+WIPBR=3,6,11** *// APN*

2.5.4.4 Displaying Physical Layer Parameters

The +WIPBR command can be used to get the current physical layer parameters such as APN, IP address, dial number.

Old interface
AT#VPHY
New interface
AT+WIPBR=3,<bid>,11 *//APN* **AT+WIPBR=3,<bid>,15** *//IP address* **AT+WIPBR=3,<bid>,2** *//Phone number to dial*

2.5.5 Bearer Configuration Management (New)

2.5.5.1 Description

The WIPSoft application provides +WIPBR command for storing configuring parameters in FLASH memory.

2.5.5.2 Syntax

```
AT+WIPBR=6,<bid>,<mode>
```

Refer to [3] for more information about the parameters and defined values and the examples.

2.6 TCP Sockets

2.6.1 Deprecated Commands

Following commands are not available in WIPSoft for TCP sockets:

- AT#DLEMODE
- AT#TCPTXDELAY

2.6.2 Socket Configuration Commands

2.6.2.1 Additional Configuration Command (New)

2.6.2.1.1 Description

The WIPSoft application provides additional options which can be used to configure the way the socket behaves.

2.6.2.1.2 Syntax

```
AT+WIPOPT=<protocol>,<idx>,2,<optnum>,<optval>
```

Refer to [3] for more information about the parameters and defined values and the examples.

2.6.2.2 Configuring TCP Parameters

The +WIPCREATE allows to configure the TCP parameters such as port number, IP address.

NOTE	The +WIPCREATE command configures and creates the socket at the same time. Configuration cannot be done separately in WIPSoft.

Old interface

```
AT#TCPORT=1,"port"
AT#TCPSERV=1,"IP address"
//In case of TCP client, these parameters are for remote server
//In case of TCP server, TCP/IP library will listen to this port and allow
//the IP address mentioned in the TCPSERV to connect
```

New interface

//For TCP client

```
AT+WIPCREATE=2,<communication index>,<peer IP>,<peer port>
```

//For TCP server

```
AT+WIPCREATE=3,<server index>,<local port>,<from idx>,<to idx>
```

//<from idx> and <to idx> indicates minimum and maximum index for spawned TCP sockets

2.6.3 Socket Creation/Termination Commands

2.6.3.1 Socket Creation

The +WIPCREATE command can be used to create TCP socket in client or server mode.

Server spawns a new socket whenever a client wants to communicate. The clients will be assigned an index based on the <from idx> and <to idx> that is specified along with the +WIPCREATE command. <from idx> indicates the minimum index that will be used between the server and the client. For the subsequent client connections the consecutive indexes till the <to idx> will be used. For example,

- Create server socket using command +WIPCREATE=3,1,80,5,10. Here the <from idx> is specified as 5 and <to idx> as 10.
- Server spawns a socket with communication index 5 when the first client connects to the server. All the communication with this client will be done through the spawned socket with the index as 5.
- Server spawns a socket with communication index 6 when the second client request for connection with the server. All the communication with this client will be done through the spawned socket with the index as 6.

Old interface
//For TCP client **AT#OTCP** *//For TCP Server* **AT#LTCPSTART**
New interface
//For TCP client, **AT+WIPCREATE=2,<communication index>,<peer IP>,<peer port>** *//For TCP server* **AT+WIPCREATE=3,<server index>,<local port>,<from idx>,<to idx>** *//From <idx> and <to idx> indicates minimum and maximum index for spawned TCP sockets*

2.6.3.2 Socket Termination

The +WIPCLOSE command can be used to terminate TCP socket in client or server mode.

Old interface

//For TCP client
`[ETX character]`
//For TCP server
`AT#LTCPSTOP`

New interface

//For TCP client
`AT+WIPCLOSE=2,<idx>`
//For TCP server
`AT+WIPCLOSE=3,<idx>`

Refer to [3] for more information about the parameters and defined values and the examples.

2.6.4 Data Transfer Command

The +WIPDATA command can be used to transfer the data to/from socket. This command switches the state of UART to data mode and allows reading/writing of data. There are 2 different modes available for data transfer and are described below:

Mode	Description
0	unmap: This parameter is used to switch the UART (mapped to continuous mode) to AT mode.
1	continuous: This parameter is used to switch the UART* to data mode. In this mode, size of the buffer need not be mentioned.

NOTE	During continuous mode, [ETX] character will terminate the session. In case [ETX] character needs to be transmitted, it should be preceded by [DLE] character. To close all sockets at once, "+++" sequence should be sent followed by +WIPCLOSE command.

Old interface
AT#OTCP CONNECT ... *//Module switches to data mode immediately after socket is created*
New interface
AT+WIPDATA=2,<idx>,<mode> *//Module can switch back and forth between AT mode and data mode as often as //wished.* *//Switches to data mode can happen on different UARTs. For instance, a socket can be created with +WIPCREATE on UART1, then the switch to data mode with +WIPDATA on UART2.* *//In continuous mode, <ETX> character must be escaped by <DLE> character.*

Refer to [3] for more information about the parameters and defined values and the examples.

2.6.5 Miscellaneous Commands

2.6.5.1 Displaying TCP Configuration Parameters

The +WIPOPT command can be used to get the current TCP configuration parameters

Old interface
AT#VTCP
New interface
//For TCP client **AT+WIPOPT=2,<idx>,1,0** *//Port* **AT+WIPOPT=2,<idx>,1,8** *//TTL* *//For TCP Server* **AT+WIPOPT=3,<idx>,1,0** *//Port* **AT+WIPOPT=3,<idx>,1**,8 *//TTL*

Refer to [3] for more information about the parameters and defined values and the examples.

2.7 UDP sockets

2.7.1 Deprecated Commands

The following commands are not available in WIPSoft for UDP sockets:

- AT#UDPTXDELAY
- AT#LUDPSTART
- AT#LUDPSTOP

2.7.2 Configuration Commands

2.7.2.1 Configuring UDP Parameters

The +WIPCREATE command can be used to configure parameters such as port number, IP address.

NOTE	The +WIPCREATE command configures and creates the socket at the same time. Configuration cannot be done separately in WIPSoft.

Old interface
AT#UDPPORT=<port number> **AT#UDPSERV=<IP address>**
New interface
AT+WIPCREATE=1,<communication index>[,<local port>][,<peer IP>, <peer port>]

2.7.3 Socket Creation/Termination Commands

2.7.3.1 Creating a UDP Socket

The +WIPCREATE command can be used to create a UDP socket.

Old interface
`AT#OUDP`
New interface
`AT+WIPCREATE=1,<communication index>[,<local port>][,<peer IP>, <peer port>]`

Refer to [3] for more information about the parameters and defined values and the examples.

2.7.3.2 Terminating a UDP Socket

The +WIPCLOSE command can be used to terminate a UDP socket

Old interface
`[ETX Character]`
New interface
`AT+WIPCLOSE=1,<idx>`

Refer to [3] for more information about the parameters and defined values and the examples.

2.7.4 Data Transfer Command

The +WIPDATA command can be used to transfer the data to/from socket. This command switches the state of UART to data mode and allows reading/writing of data.

UDP is a connectionless protocol and hence there is no way to detect or cause a shutdown. However, an [ETX] character is used to mark the boundaries of datagrams.

All data written on an UDP socket is collected till an [ETX] character is encountered or the maximum size of the datagram[1] is reached and will be sent as a single datagram. Similarly when reading data, all data will be read till an [ETX] character is encountered which indicates the end of the datagram.

There are 2 different modes available for data transfer and are described below:

Mode	Description
0	unmap: This parameter is used to switch the UART (mapped to continuous mode) to AT mode.
1	continuous: This parameter is used to switch the UART* to data mode. In this mode, size of the buffer need not be mentioned.

NOTE	During continuous mode, [ETX] character will terminate the session. In case [ETX] character needs to be transmitted, it should be preceded by [DLE] character. To close all sockets at once, "+++" sequence should be sent followed by +WIPCLOSE command. The UART switches back to AT mode due to "+++" sequence or +WIPDATA=1,x,0 command.

[1] *Maximum size of an UDP datagram has been fixed to 5840 Bytes. This limit is an arbitrary one. Nevertheless, note that smaller the datagram is the surer it will reach the aimed destination. Note that UDP is not a reliable transport layer.*

Old interface
AT#OUDP CONNECT ... *//Module switches to data mode immediately after socket is created*
New interface
AT+WIPDATA=1,<idx>,<mode> *//Module switches to data mode manually*

Refer to [3] for more information about the parameters and defined values and the examples.

2.7.5 Miscellaneous Commands

2.7.5.1 Displaying UDP Parameters

The +WIPOPT command can be used to get the current UDP parameters.

Old interface
AT#VUDP
New interface
AT+WIPOPT=1,<idx>,1,0 *//Port* **AT+WIPOPT=1,<idx>,1,8** *//TTL*

Refer to [3] for more information about the parameters and defined values and the examples.

2.8 FTP Service

2.8.1 Configuration Commands

2.8.1.1 Configuring FTP Server Parameters

The +WIPCREATE command can be used to configure parameters such as port number, IP address of FTP server

NOTE	The +WIPCREATE command configures and creates the FTP session at the same time. Configuration cannot be done separately in WIPSoft

Old interface
AT#FTPPORT=<port number> **AT#FTPUN=<username>** **AT#FTPPW=<password>** **AT#FTPSERV=<IP address>**
New interface
AT+WIPCREATE=4,<index>,<server>[,<peer_port>],<username>, <password>[,<account>]

Refer to [3] for more information about the parameters and defined values and the examples.

2.8.1.2 Configuring FTP Transfer Parameters

The +WIPOPT command can be used to configure FTP transfer related parameters such as mode of transfer.

Old interface
AT#FTPTYPE=<type of data transfer> **AT#FTPMODE=<mode of data transfer>**
New interface
AT+WIPOPT=4,<idx>,2,40,<optval> **AT+WIPOPT=4,<idx>,2,41,<optval>**

Refer to [3] for more information about the parameters and defined values and the examples.

2.8.1.3 Configuring Parameters Related to File Upload

The +WIPFILE command can be used to set the file name to be uploaded.

NOTE	The +WIPFILE command sets the file name to be uploaded and uploads the file at the same time. The file name cannot be set separately in WIPSoft.

Old interface
AT#FTPPUTFILENAME=<filename> **AT#FTPPUTPATH=<path of file>**
New interface
AT+WIPFILE=4,<idx>,2,<filename> *//Filename contains both path as well as file name*

2.8.1.4 Configuring Parameters Related to File Download

The +WIPFILE command can be used to set the file name to be downloaded.

NOTE	The +WIPFILE command sets the file name to be downloaded and downloads the file at the same time. The file name cannot be set separately in WIPSoft

Old interface
AT#FTPGETFILENAME=<filename> **AT#FTPGETPATH=<path of file>**
New interface
AT+WIPFILE=4,<idx>,1,<filename> *//Filename contains both path as well as file name*

2.8.2 Uploading a File

The +WIPFILE command can be used to upload a file to the FTP server. The <ETX> character indicates end of the data in the file that is being transferred.

Old interface
AT#FTPPUT <data>
New interface
AT+WIPFILE=4,<idx>,2,<filename> <data>

NOTE	In case [ETX] character needs to be transmitted, it should be preceded by [DLE] character. If a "+++" escape sequence is sent during file transfer, it is interpreted as an [ETX] character.

Refer to [3] for more information about the parameters and defined values and the examples.

2.8.3 Downloading a File

The +WIPFILE command can be used to download a file from the FTP server. The <ETX> character indicates end of the data in the file that is being transferred.

Old interface
AT#FTPGET <data>
New interface
AT+WIPFILE=4,<idx>,1,<filename> <data>

Refer to [3] for more information about the parameters and defined values and the examples.

NOTE	In case [ETX] character needs to be transmitted, it should be preceded by [DLE] character. If a "+++" escape sequence is sent during file transfer, it is interpreted as an [ETX] character.

2.8.4 Miscellaneous Commands

2.8.4.1 Displaying FTP Related Parameters

The +WIPOPT command can be used to display the parameters related to FTP.

Old interface
AT#VFTP
New interface
AT+WIPOPT=4,<idx>,1,40 **AT+WIPOPT=4,<idx>,1,41**

2.8.5 Closing a FTP Connection

The +WIPCLOSE command can be used to close the FTP session.

Old interface
//Session closes automatically after the file is downloaded, in case //of upload, the session is closed after data transfer
New interface
AT+WIPCLOSE=4,<idx>

Refer to [3] for more information about the parameters and defined values and the examples.

2.9 PING Service

2.9.1 Deprecated Commands

The following command is not available in WIPSoft for UDP sockets:

- AT#VPING

2.9.2 Configuration Command

2.9.2.1 Configuring PING Related Parameters

The +WIPPING can be used to configure the PING related parameters.

NOTE	The +WIPPING command configures parameters and creates the PING session at the same time. Configuration cannot be done separately in WIPSoft

Old interface
AT#PINGDELAY=<interval> **AT#PINGNUM=<repeat>** **AT#PINGREMOTE=<host>**
New interface
AT+WIPPING=<host>[,<repeat>,<interval>,[<timeout>,[<nwrite>,[ttl>]]]]

2.9.3 PING Session Creation Command

2.9.3.1 Creating a PING Session

The +WIPPING can be used to ping a remote server.

Old interface
AT#PING
New interface
AT+WIPPING=<host>[,<repeat>,<interval>,[<timeout>,[<nwrite>,[ttl>]]]]

Refer to [3] for more information about the parameters and defined values and the examples.

2.10 SMTP/POP3 Service

2.10.1 SMTP/POP3 Service Commands in eDSoft

The eDSoft application provides commands which can be used to send/receive emails using SMTP/POP3 protocol. Following commands are present to send/receive emails.

- AT#SENDERNAME
- AT#SENDERADDR
- AT#CCREC1/CCREC2/CCREC3
- AT#DOMAIN
- AT#REC1,REC1ADD/REC2,REC2ADD/REC3,REC3ADD
- AT#SUBJ1/SUBJ2/SUBJ3
- AT#BODY1/BODY2/BODY3
- AT#POP3HEADERMODE
- AT#POP3PORT
- AT#POP3PW
- AT#POP3SERV
- AT#POP3UN
- AT#SMTPPORT
- AT#SMTPPW
- AT#SMTPSERV
- AT#SMTPUN
- AT#DNSSERV1
- AT#DNSSERV2
- AT#GETMAIL
- AT#SENDMAIL1/SENDMAIL2/SENDMAIL3
- AT#PUTMAIL
- AT#VMAIL1/VMAIL2/VMAIL3
- AT#VPOP3
- AT#VSMTP
- AT#VDNS

2.10.2 SMTP/POP3 Service Commands in WIPSoft

Currently, the WIPSoft application does not support the SMTP/POP3 service commands. These commands will be added in future releases of WIPSoft application.

2.11 Miscellaneous Commands

2.11.1 Deprecated Commands

The following miscellaneous commands are not available in the WIPSoft:

- AT#VSTATE
- AT#VALL
- AT#DELFLASH

2.11.2 IP Stack Version Information

The +WIPCFG command is used to get the version information for the TCP/IP library.

Old interface
AT#VVERSION
New interface
AT+WIPCFG=3

Make it wireless

WAVECOM S.A. - 3 esplanade du Foncet - 92442 Issy-les-Moulineaux Cedex - France - Tel: +33(0)1 46 29 08 00 - Fax: +33(0)1 46 29 08 08
Wavecom, Inc. - 4810 Eastgate Mall - Second Floor - San Diego, CA 92121 - USA - Tel: +1 858 362 0101 - Fax: +1 858 558 5485
WAVECOM Asia Pacific Ltd. - 4/F, Shui On Centre - 6/8 Harbour Road - Hong Kong - Tel: +852 2824 0254 - Fax: +852 2824 025

www.wavecom.com