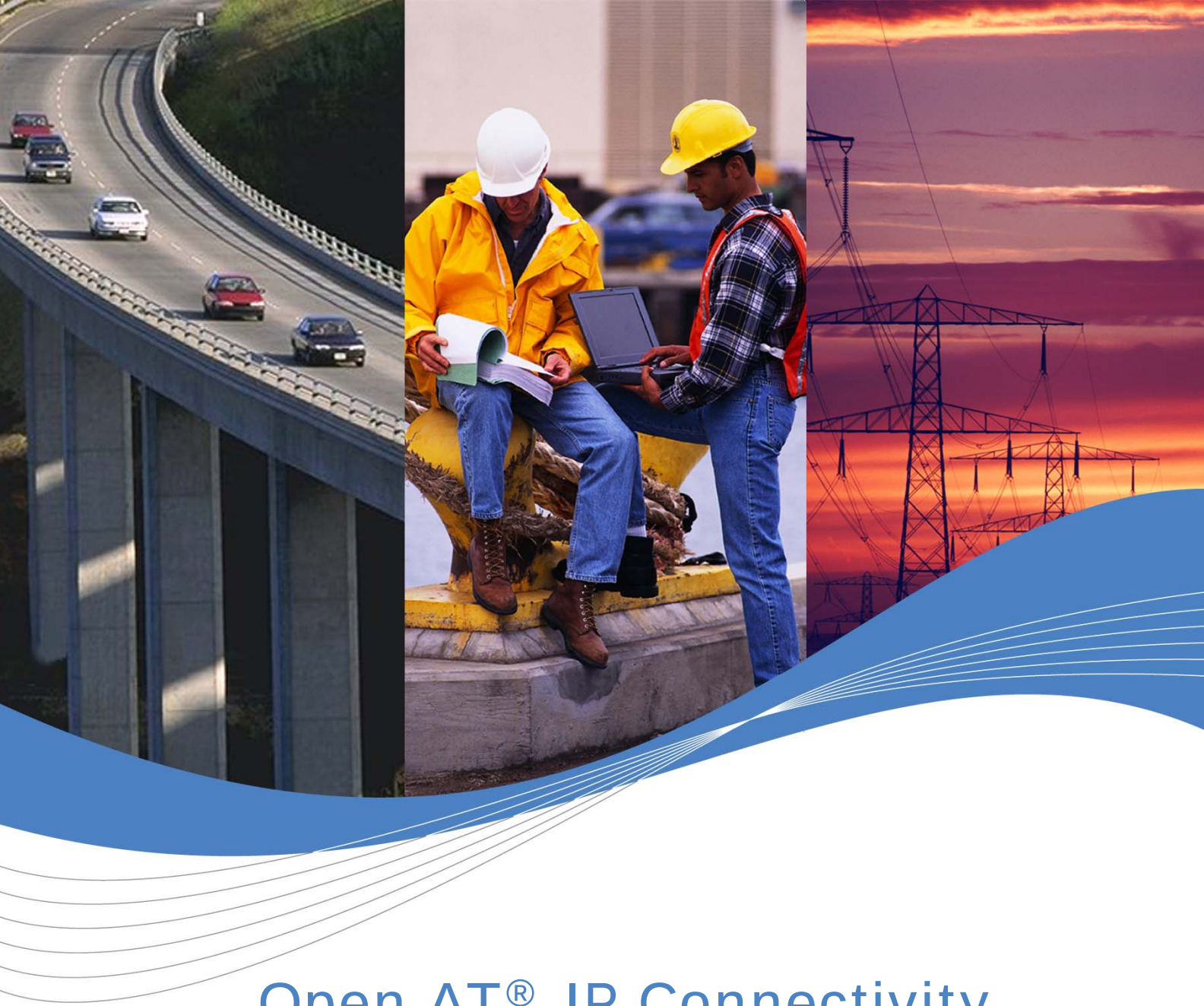


Open AT® IP Connectivity Development Guide (eDLib V3.03)

Revision: 005
Date: December 2005

wavecom
Make it wireless

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

Open AT® IP Connectivity Development Guide (eDLib V3.03)

Revision: **005**

Date: **14th December 2005**

Reference: **WM_ASW_OAT_UGD_012**

Document History

Revision	Date	Document history	
001	04/11/03	First edition	
002	05/01/04	Added UDP socket & PING services. Updates in SMTP service to support the multi-recipient feature and the authentication feature. Added new error code ED_ERR_GPRS_SESSION_LOST for the responsecbk() of some services and ED_ERR_SERVICE_ALREADY_RUNNING for application callback (WMUX security)	
003	Jan 2005	Updates for Open AT® V3.00	
004	May 2005	Updates for eDlib V3.02	
005	December 14, 2005	Updates for eDlib V3.03	

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

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

This manual is copyrighted by WAVECOM with all rights reserved. No part of this manual may be reproduced in any form without the prior written permission of WAVECOM.

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

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

Overview

This document describes the interfaces of the IP Connectivity library for the WAVECOM Open AT® software.

It applies to the Open AT® **library eDLib V3.03 and upper until further notice.**

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

Contents

DOCUMENT HISTORY 2

TRADEMARKS 3

COPYRIGHT 3

OVERVIEW 4

CONTENTS 5

1 DESCRIPTION 9

2 OPEN AT® IP CONNECTIVITY LIBRARY 10

2.1 Required Header Files. 10

2.2 Initialization 10

2.3 Dialing Service 11

2.3.1 Incoming Call Management 11

2.3.1.1 Required Header Files. 11

2.3.1.2 The ed_answerSetupParams_t type 11

2.3.1.3 The ed_AnswerSetConfig function 15

2.3.1.4 The ed_AnswerGetConfig function 15

2.3.1.5 The ed_Accept function 16

2.3.2 Dial Up Management 17

2.3.2.1 Required Header Files 17

2.3.2.2 The ed_dialupSetupParams_t type 17

2.3.2.3 The ed_DialupSetConfig function 19

2.3.2.4 The ed_DialupGetConfig function 19

2.3.2.5 The ed_DialupConnectionStart function 20

2.3.2.6 The ed_DialupConnectionStop function 22

2.3.2.7 The ed_DialupGetIPInformations function 23

2.3.2.8 The ed_DialupGetGeneralInformations function 24

2.3.3 PPP Management 25

2.3.3.1 Required Header Files 25

2.3.3.2 The ed_pppSetupParams_t type 25

2.3.3.3 The ed_PPPSetConfig function 27

2.3.3.4 The ed_PPPGetConfig function 28

2.3.4 DNS Management 28

2.3.4.1 Required Header Files 28

2.3.4.2 The ed_dnsSetupParams_t type 28

2.3.4.3 The ed_DNSSetConfig function 29

2.3.4.4 The ed_DNSGetConfig function 29

2.3.5 GPRS Management 30

2.3.5.1 Required Header Files 30

2.3.5.2 The ed_gprsSetupParams_t type 30

2.3.5.3 The ed_GprsSetConfig function 31

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.3.5.4 The ed_GprsGetConfig function 32

2.4 SMTP/POP3 e-mail Service 33
2.4.1 SMTP e-mail services 33
2.4.1.1 Required Header Files. 33
2.4.1.2 The ed_EmailSetupParams_t type 33
2.4.1.3 The ed_EmailSetConfig function 35
2.4.1.4 The ed_EmailGetConfig function 35
2.4.1.5 The ed_EmailAddrParams_t type. 36
2.4.1.6 The ed_EmailAddrSet function. 36
2.4.1.7 The ed_EmailAddrGet function. 37
2.4.1.8 The ed_SMTPSetupParams_t type 38
2.4.1.9 The ed_SMTPSetConfig function 40
2.4.1.10 The ed_SMTPGetConfig function 40
2.4.1.11 The ed_SendMail function 41
2.4.1.12 The ed_PutMail function 42
2.4.2 POP3 service 44
2.4.2.1 Required Header Files. 44
2.4.2.2 The ed_POP3SetupParams_t type 44
2.4.2.3 The ed_POP3SetConfig function 46
2.4.2.4 The ed_POP3GetConfig function 47
2.4.2.5 The ed_GetMail function 47

2.5 FTP Service 49
2.5.1 Required Header Files. 49
2.5.2 The ed_FTPSetupParams_t type 49
2.5.3 The ed_FTPSetConfig function 52
2.5.3.1 Prototype 52
2.5.3.2 Parameters 52
2.5.3.3 Returned values 52
2.5.4 The ed_FTPGetConfig function 52
2.5.4.1 Prototype 52
2.5.4.2 Parameters 52
2.5.4.3 Returned values 52
2.5.5 The ed_FTPGetFileParams_t type 53
2.5.6 The ed_FTPGetFileSetConfig function 54
2.5.6.1 Prototype 54
2.5.6.2 Parameters 54
2.5.6.3 Returned values 54
2.5.7 The ed_FTPGetFileGetConfig function 54
2.5.7.1 Prototype 54
2.5.7.2 Parameters 54
2.5.7.3 Returned values 54
2.5.8 The ed_FTPPutFileParams_t type 55
2.5.9 The ed_FTPPutFileSetConfig function 56
2.5.9.1 Prototype 56
2.5.9.2 Parameters 56
2.5.9.3 Returned values 56
2.5.10 The ed_FTPPutFileGetConfig function 56
2.5.10.1 Prototype 56
2.5.10.2 Parameters 56
2.5.10.3 Returned values 56
2.5.11 The ed_FTPGet function 57
2.5.11.1 Prototype 57
2.5.11.2 Parameters 57
2.5.11.3 Returned values 58

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.5.12 The ed_FTPPut function 59
2.5.13 Prototype 59
2.5.13.1 Parameters 59
2.5.13.2 Returned values 60
2.6 TCP SOCKET Service 61
2.6.1 Required Header Files. 61
2.6.2 The ed_SocketSetupParams_t type 61
2.6.3 Socket Id type 62
2.6.4 The ed_SocketSetConfig function 63
2.6.4.1 Prototype 63
2.6.4.2 Parameters 63
2.6.4.3 Returned values 63
2.6.5 The ed_SocketGetConfig function 64
2.6.5.1 Prototype 64
2.6.5.2 Parameters 64
2.6.5.3 Returned values 64
2.6.6 The ed_SocketTCPStart function 65
2.6.6.1 Prototype 65
2.6.6.2 Parameters 65
2.6.6.3 Returned values 67
2.6.7 The ed_SocketLTCPStop function 67
2.6.7.1 Prototype 67
2.6.7.2 Parameters 68
2.6.7.3 Returned values 68
2.7 UDP SOCKET Service 68
2.7.1 Required Header Files. 68
2.7.2 The ed_UdpSocketSetupParams_t type 68
2.7.3 The ed_UdpSocketSetConfig function 70
2.7.3.1 Prototype 70
2.7.3.2 Parameters 70
2.7.3.3 Returned values 70
2.7.4 The ed_UdpSocketGetConfig function 71
2.7.4.1 Prototype 71
2.7.4.2 Parameters 71
2.7.4.3 Returned values 71
2.7.5 The ed_SocketUDPStart function 71
2.7.5.1 Prototype 72
2.7.5.2 Parameters 72
2.7.5.3 Returned values 73
2.7.6 The ed_SocketLUDPStop function 74
2.7.6.1 Prototype 74
2.7.6.2 Parameters 74
2.7.6.3 Returned values 74
2.8 PING Service 75
2.8.1 Required Header Files. 75
2.8.2 The ed_PINGSetupParams_t type 75
2.8.3 The ed_PINGSetConfig function 76
2.8.3.1 Prototype 76
2.8.3.2 Parameters 76
2.8.3.3 Returned values 76
2.8.4 The ed_PINGGetConfig function 76
2.8.4.1 Prototype 76
2.8.4.2 Parameters 76
2.8.4.3 Returned values 76

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.8.5 The ed_PingEchoRequest function 77
2.8.5.1 Prototype 77
2.8.5.2 Parameters 77
2.8.5.3 Returned values 77
2.8.6 The ed_PingReset function 78
2.8.6.1 Prototype 78
2.8.6.2 Returned values 78
2.9 Data Flows Management 78
2.9.1 Function ed_SendData 78
2.9.1.1 Prototype 78
2.9.1.2 Parameters 78
2.9.1.3 Returned values 79
2.9.2 Function ed_SendDataExt 79

3 LIST OF ERROR CODES AND REPLY CODES 81

4 APPENDIX: OPEN AT® RESOURCES 86
4.1 Flash objects 86
4.2 Timers 86
4.3 RAM and ROM occupation 86

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

1 Description

The aim of this document is to describe the API dedicated to the implementation of Wavecom IP connectivity library on Wavecom modules WISMO Q24x6B, Q24x6A and Q2501B.

This library offers to the Open AT® application developers a direct access to the following Internet protocols:

Entry level TCP/IP configuration :

- TCP socket,
- DNS (domain name service)
- UDP Socket
- PING Service

High level TCP/IP configuration :

- All Entry level TCP/IP configuration services
- POP3 (e-mail),
- SMTP (e-mail),
- FTP (file transfer protocol),

Open AT® application has to be developed using Wavecom ADL environment and has to be linked with eDLib V3.03 library file.

NOTE: to use the IP connectivity library services, the GPRS or CSD connection must be done through the IP connectivity library dialing service.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2 Open AT® IP connectivity library

The Open AT IP Connectivity library allows only one IP application (FTP, SMTP, POP3 or socket TCP, socketUDP, PING) working at a time.

2.1 Required Header Files.

The interface header files shared by all IP connectivity services are:
ed_common.h

2.2 Initialization

This function must be called in the **`adl_main`** of the client application.

This function initializes the IP connectivity library.
It loads IP connectivity parameters from EEPROM: these are default parameters or parameters saved in EEPROM.

Notes: If the client application wants to initialize PIN code using ADL function **`adl_simSubscribe`**, **`ed_Init`** must be called **after** **`adl_simSubscribe`**.

- **Prototype**

```
s8 ed_Init(void);
```

- **Parameters**
  None

- **Returned values**

This function returns ED_ENTRY_LEVEL_TCP_IP or ED_HIGH_LEVEL_INTERNET on success, or a negative error value. Possible error values are:

- ED_ERR_TCPIP_ALREADYUSED
- ED_ERR_WM_UNABLE_SUBS_EVENT
- ED_ERR_PARAM_READ_E2P
- ED_ERR_STACK_INTERNAL

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.3 Dialing Service

IP connectivity library provides this service to handle GSM and GPRSdialing related events, and to setup dialing parameters.

2.3.1 Incoming Call Management

This set of parameters defines how the TCP/IP stack will behave when receiving an incoming call.

2.3.1.1 Required Header Files.

The interface header file for the Incoming Call Management service is:

ed_answer.h

2.3.1.2 The ed_answerSetupParams_t type

Structure containing the parameters about the GSM incoming call management.

```
#define ANSWER_STRMAXSIZE       64


typedef struct
{
    u8 AnswerMode;
    u8 CallbackTimer;
    ascii CallScreenNum[ANSWER_STRMAXSIZE+1];
    u8 RingCount;
    ed_ResponseCbk_f ResponseCbk;
}ed_answerSetupParams_t;
```


This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

- **ANSWERMODE:**
  This parameter defines how the TCP/IP stack will behave when receiving an incoming call in accordance with the following values:

Value	Description
0 (default)	(Ignore) ignores the incoming call. In this case, it is the responsibility of the application to accept/refuse the incoming call by issuing the ed_Accept() function (see §2.3.1.5).
1	(Automatic Answer) The TCP/IP stack goes off hook and accepts the incoming call. As described below, the calling number must match the one specified in the CallScreenNum parameter. The RingCount parameter should be > 0
2	(Static Callback) The TCP/IP stack ignores the incoming call and then automatically dials DIALN1 or DIALN2 (see §2.3.2.2) by issuing the ed_DialupConnectionStart() function (see §2.3.2.5). As described below, the calling number must match the one specified in the CallScreenNum parameter. The RingCount parameter should be > 0. It also depends on the format of the caller phone number.
3	(Dynamic Callback) The TCP/IP stack ignores the incoming call and then automatically dials the calling number by issuing the ed_DialupConnectionStart() function (see §2.3.2.5). For this feature, the CallerID service is mandatory. As described below, the calling number must match the one specified in the CallScreenNum parameter. Ensure that the calling number is compatible with the local telephone network (extension issues).

  Note: The AnswerMode parameter must be configured in accordance with ATS0 configuration for not interacting.

- **CALLBACKTIMER:**
  This parameter defines the number of seconds (between 2 and 255 inclusive) the TCP/IP stack will wait before an automatic callback operation occurs after receiving an incoming call. It only applies when the ANSWERMODE parameter is set to an automatic callback mode (value>1).

  This timer starts after the end of the ringing signal.

  By default this parameter is set to 2.

- **CALLSCREENNUM:**
  When receiving an incoming call, the caller identification (Caller ID) service allows the TCP/IP stack to identify the phone number of the remote caller. This information is helpful in preventing unauthorized callers to trigger actions on the TCP/IP stack.

  This parameter allows the user to filter the incoming calls when the ANSWERMODE parameter is set to an automatic mode (value>0). This filtering does not apply when the ANSWERMODE parameter is set to 0, in this case it is the hosts responsibility to accept or reject the incoming call.

  If an incoming phone number is unauthorized, the TCP/IP stack will ignore it.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

This parameter is set in accordance with the following values:

0 (zero): No remote caller authorized (default value)

** (all)*: No filtering is applied on incoming calls. All the remote phone numbers are authorized. This value must be set when wanting to receive incoming calls while the Caller ID service is not available.

Decimal phone number: Only the phone number configured here before is authorized for incoming calls.

By default this parameter is set to 0.

Note: this parameter must be a NULL terminated string.

- **RINGCOUNT:**
  This parameter defines the number of rings (range [0 ; 15]) that will be waited before an automatic operation occurs when receiving an incoming call.

  This parameter only applies when the ANSWERMODE parameter is set to an automatic mode (value>0).

  If the ANSWERMODE parameter is used (value different from 0), the RINGCOUNT value should be >0 to use the feature.

  By default this parameter is set to 0.

  Note: The RINGCOUNT must be configured in accordance with ATS0 configuration for not interacting.

- **RESPONSECBK:**
  ResponseCbk is called by the IP connectivity library when incoming call is automatically accepted, when the IP session is opened or if the accept action fails.

  This callback function is defined as below:

```
typedef void (* ed_ ResponseCbk_f) (s32 ResponseCode,
                                    TeDHandle id));
```

  TeDHandle id: always set to ED_ID_INCOMING

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

ResponseCode values:

- ED_INFO_CONNECT

  Modem speed negotiated between both side. It can be from the V32bis to V.90 depending on the software and also on the conditions. If the TCP/IP stack is configured for modem only operation (PPPMODE parameter, see §2.3.3.2) there is no more return codes.

- ED_OK_PPP
- ED_ERR_BUSY
- ED_ERR_NO_ANSWER
- ED_ERR_NO_CARRIER
- ED_ERR_PPP_ERROR
- ED_ERR_PPP_TIMEOUT
- ED_ERR_INVALID_EVENT
- ED_ERR_PHY_TIMEOUT
- ED_ERR_NETWORK_KO
- ED_ERR_WM_FLOW_OPEN
- ED_ERR_WM_PARSER_FLOW
- ED_ERR_CLI_INVALID

See §3 - List of error codes and reply codes for more information on returned codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.3.1.3 The ed_AnswerSetConfig function

This function sets up some specific parameters about the incoming call management.

- **Prototype**

```
s8 ed_AnswerSetConfig(ed_answerSetupParams_t  *Params);
```

- **Parameters**

Params:
Structure containing the parameters to set up. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:

  - ED_ERR_NOT_INIT
  - ED_ERR_PARAM_BAD_VALUE
  - ED_ERR_PARAM_TOO_LONG
  - ED_ERR_PARAM_WRITE_E2P
  - ED_ERR_INVALID_ARG
  - ED_ERR_SERVICE_ALREADY_RUNNING

See §3 - List of error codes and reply codes for more information on error codes.

2.3.1.4 The ed_AnswerGetConfig function

This function gets up some specific parameters about the incoming call management.

- **Prototype**

```
s8 ed_AnswerGetConfig(ed_answerSetupParams_t  *Params);
```

- **Parameters**

Params:
Structure containing the parameters set. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:

  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.3.1.5 The ed_Accept function

This command directs the TCP/IP stack to answer an incoming call.

When the incoming call is received, "RING" messages are sent to the customer application (if it did subscribe to the corresponding service). Depending on the value of the ANSWERMODE parameter(see §2.3.1.2) the TCP/IP stack may answer automatically or not.

If ANSWERMODE(see §2.3.1.2) is set to 0, it is the application that is responsible for answering the incoming call with this function.

- **Prototype**

```
s8 ed_Accept (ed_ResponseCbk_f ResponseCbk);
```

- **Parameters**

  ResponseCbk:

  ResponseCbk is called by the IP connectivity library when the IP session is opened or if the accept action fails.

  This callback function is defined as below:

```
typedef void (* ed_ ResponseCbk_f) (s32 ResponseCode,
                                    TeDHandle id);
```

TeDHandle id: always set to ED_ID_ACCEPT

ResponseCode values:

- ED_INFO_CONNECT

  Modem speed is negotiated between both sides. It can be from the V32bis to V.90 depending on the software and also on the conditions. If the TCP/IP stack is configured for modem only operation (PPPMODE parameter, see §2.3.3.2) there is no more return codes.

- ED_OK_PPP
- ED_ERR_BUSY
- ED_ERR_NO_ANSWER
- ED_ERR_NO_CARRIER
- ED_ERR_PPP_ERROR
- ED_ERR_PPP_TIMEOUT
- ED_ERR_INVALID_EVENT
- ED_ERR_PHY_TIMEOUT
- ED_ERR_NETWORK_KO
- ED_ERR_WM_FLOW_OPEN
- ED_ERR_WM_PARSER_FLOW

- **Returned values**

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG
- ED_ERR_MODEM_RUNING
- ED_ERR_INVALID_RING_TYPE

See §3 - List of error codes and reply codes for more information on returned values and error codes.

NOTE: Once the physical layer is up, the TCP/IP stack runs the applicable protocol as specified in the PPPMODE parameter (see §2.3.3.2).

The ed_DialupConnectionStop() function (see §2.3.2.6) allows to on hook and close the connection.

2.3.2 Dial Up Management

2.3.2.1 Required Header Files

The interface header file for the Dial Up Management service is:

ed_dialup.h

2.3.2.2 The ed_dialupSetupParams_t type

Structure containing the parameters about the GSM dial up management.

```
#define DIALUP_STRMAXSIZE  64


typedef struct
{
    ascii DIALN1[DIALUP_STRMAXSIZE+1];
    ascii DIALN2[DIALUP_STRMAXSIZE+1];
    u8 DIALSELECT;
    ascii ISPUN[DIALUP_STRMAXSIZE+1];
    ascii ISPPW[DIALUP_STRMAXSIZE+1];
    u8 REDIALCOUNT;
    u8 REDIALDELAY;
    u8 PHYTIMEOUT;
}ed_dialupSetupParams_t;
```

- DIALN1 and DIALN2:
  Primary and secondary dial-up phone number to connect with the local ISP. Length depends on country.

  By default these parameters are set to empty strings.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

- DIALSELECT:
  The value of this parameter determines the number called to establish an Internet connection. It configures the TCP/IP stack software to use the primary dial-up number or the secondary dial-up number.

  By default this parameter is set to 1.

- ISPUN:
  User name of the ISP account in ASCII text string up to 64 characters.
  When communication is initiated and once the physical (modem) connection has been established with the ISP, the TCP/IP stack software must provide the ISP with the user name associated with the account to be used.

  By default this parameter is set to empty string.

- ISPPW:
  Password for the ISP account in ASCII text string up to 64 characters.

  When communication is initiated and once the physical (modem) connection has been established with the ISP, the TCP/IP stack software must provide the ISP with the password associated with the account to be used.

  By default this parameter is set to empty string.

- REDIALCOUNT:
  Indicates how many unsuccessful connection (between 0 and 14, inclusive) attempts the TCP/IP stack software will try before terminating the connection attempt.
  If the value is set to 0, the TCP/IP stack software will not make any call retry.

  By default this parameter is set to 5.

- REDIALDELAY:
  This parameter controls how long (in seconds between 5 and 14 inclusive), if any, it will take between each call retry.
  If this parameter is configured to zero, the TCP/IP stack software will attempt another connection immediately after terminating the previous unsuccessful attempt.

  By default this parameter is set to 5.

- PHYTIMEOUT:
  This parameter is used by the TCP/IP stack software in order to terminate connections to the telephone line when a long period elapses without activity. “Without activity” is defined as a period (between 1 and 255 minutes) when no data is transferred between the Internet and the TCP/IP stack software. This timer prevents the telephone line from being tied up if for any reason some part of the system becomes stuck.

  By default this parameter is set to 15.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.3.2.3 The ed_DialupSetConfig function

This function sets up some specific parameters about the dial up management.

- **Prototype**

```
s8 ed_DialupSetConfig(ed_dialupSetupParams_t  *Params);
```

- **Parameters**

  Params:
  Structure containing the parameters to set up. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NOT_INIT
  - ED_ERR_PARAM_BAD_VALUE
  - ED_ERR_PARAM_TOO_LONG
  - ED_ERR_PARAM_WRITE_E2P
  - ED_ERR_INVALID_ARG
  - ED_ERR_SERVICE_ALREADY_RUNNING

See §3 - List of error codes and reply codes for more information on error codes.

2.3.2.4 The ed_DialupGetConfig function

This function gets up some specific parameters about the dial up management.

- **Prototype**

```
s8 ed_DialupGetConfig(ed_dialupSetupParams_t  *Params);
```

- **Parameters**

  Params:
Structure containing the parameters set. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.3.2.5 The ed_DialupConnectionStart function

Important note: GPRS attachment must be performed before calling this function.

This function directs the TCP/IP stack to dial out and establish the connection.

Upon receiving this signal, the TCP/IP stack automatically initiates a complete connection session according to the GPRSMODE parameter (see §2.3.5.2), selecting the GSM or GPRS mode.

- In GSM mode, the TCP/IP stack will dial the number according to the Dial Option parameter (DIALN1 or DIALN2 depending on DIALSELECT (see §2.3.2.2)). If an error occurs, the TCP/IP stack automatically attempts according to the REDIALCOUNT parameter (see §2.3.2.2).

Once the physical layer is up, the TCP/IP stack runs the applicable protocol as specified in the PPPMODE parameter (see §2.3.3.2).

- In GPRS mode, the TCP/IP stack will establish a GPRS session with the APN using APNUN, APNPW, CID parameter (see §2.3.5.2). Once the GPRS link is up, the product is connected to the Internet.

- **Prototype**

```
s8 ed_DialupConnectionStart (ed_ResponseCbk_f ResponseCbk);
```

- **Parameters**

ResponseCbk:

ResponseCbk is called by the IP connectivity library when the connection is established or if the connection fails.

This callback function is defined as below:

```
typedef void (* ed_ ResponseCbk_f) (s32 ResponseCode,
                                    TeDHandle id);
```

TeDHandle id: always set to ED_ID_DIALUP_START

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

ResponseCode values:

- ED_INFO_CONNECT

  Modem speed negotiated between both side.If the TCP/IP stack is configured for modem only operation (PPPMODE parameter, see §2.3.3.2) there is no more return codes.

- ED_OK_PPP
- ED_ERR_BUSY

  A busy signal is detected on the remote site. TCP/IP stack will wait REDIALDELAY seconds (see §2.3.2.2) and then dials again. This re-dialing will continue until success or until the number of call retries, defined in parameter REDIALCOUNT (see §2.3.2.2), has been reached.

- ED_ERR_NO_ANSWER

  There is no response from the remote site. TCP/IP stack will wait REDIALDELAY seconds (see §2.3.2.2) and then dials again. This re-dialing will continue until success or until the number of call retries, defined in parameter REDIALCOUNT (see §2.3.2.2), has been reached.

- ED_ERR_NO_CARRIER

  The modem handshaking process with the remote host is interrupted or unsuccessful. TCP/IP stack will wait REDIALDELAY seconds (see §2.3.2.2) and then dials again. This re-dialing will continue until success or until the number of call retries, defined in parameter REDIALCOUNT (see §2.3.2.2), has been reached.

- ED_ERR_PPP_ERROR
- ED_ERR_PPP_TIMEOUT
- ED_OK_GPRS_SESSION_SET
- ED_ERR_INVALID_EVENT
- ED_ERR_PHY_TIMEOUT
- ED_ERR_NETWORK_KO
- ED_ERR_SIM_REMOVED
- ED_ERR_WM_FLOW_OPEN
- ED_ERR_WM_PARSER_FLOW
- ED_ERR_WM_GPRS_OPEN
- ED_ERR_GPRS_ABORTED
- ED_ERR_WM_GPRS_IPCP
- ED_ERR_GPRS_SESSION_LOST

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NOT_INIT
  - ED_ERR_PIN_NOT_ENTERED
  - ED_ERR_NETWORK_KO
  - ED_ERR_INVALID_ARG
  - ED_ERR_NO_GPRS
  - ED_ERR_MODEM_RUNNING

See §3 - List of error codes and reply codes for more information on error codes.

NOTE: The ed_DialupConnectionStop() function (see §2.3.2.6) allows to on hook and close the connection.

2.3.2.6 The ed_DialupConnectionStop function

This function directs the TCP/IP stack to end a GPRS or GSM communication previously established with a START command.

- **Prototype**

```
s8 ed_DialupConnectionStop (ed_ResponseCbk_f ResponseCbk);
```

- **Parameters**

ResponseCbk:

ResponseCbk is called by the IP connectivity library when the connection is closed or if an error occurs.

This callback function is defined as below:

```
typedef void (* ed_ ResponseCbk_f) (s32 ResponseCode,
                                    TeDHandle id);
```

TeDHandle id: always set to ED_ID_DIALUP_STOP.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

ResponseCode values:

- ED_OK_ON_HOOK
- ED_ERR_ON_HOOK
- ED_ERR_WM_FLOW_CLOSE
- ED_ERR_NETWORK_KO
- ED_ERR_SIM_REMOVED
- ED_ERR_NO_CARRIER

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG
  - ED_ERR_PHY_NOT_ACTIVATED

See §3 - List of error codes and reply codes for more information on error codes.

2.3.2.7 The ed_DialupGetIPInformations function

This function gets information about the activated IP connection.

- **Prototype**

```
s8 adl_DialupGetIpInformations (ed_dialupIPInfos_t * Infos);
```

- **Parameters**

  Infos:
  Structure containing the data of the activated IP connection using the following type:

```
typedef struct
{
  u32 LocalIP; // Local IP address of the MS (only if is activated, else 0)
  u32 DistantIP; // Distant IP address (only if is activated, else 0)
  u32 Gateway; // Gateway IP address (only if is activated, else 0)
} ed_dialupIPInfos_t;
```

- **Returned values**

  This function returns 0 on success, or a negative error value.

  Possible error values are:
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG
  - ED_ERR_PHY_NOT_ACTIVATED

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.3.2.8 The ed_DialupGetGeneralInformations function

This function gets general information about the TCP/IP stack.

- **Prototype**

```
s8 adl_DialupGetGeneralInformations (ed_dialupGenInfos_t * Infos);
```

- **Parameters**

  Infos:
  Structure containing the data using the following type:

```
typedef struct
{
    ed_state_e STATE;  // current status of the IP connectivity library.
    ascii VERSION[100]; // current version of the IP connectivity library
} ed_dialupGenInfos_t;

typedef enum
{
      ED_INIT,
      ED_NO_SERVICE,
      ED_IDLE,
      ED_STARTING,
      ED_DIALING,
      ED_AUTHENTICATING,
      ED_CONNECTED,
      ED_DISCONNECTING
} ed_state_e
```

- **Returned values**

  This function returns 0 on success, or a negative error value.

  Possible error values are:
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.3.3 PPP Management

2.3.3.1 Required Header Files

The interface header file for the PPP management service is:

ed_ppp.h

2.3.3.2 The ed_pppSetupParams_t type

Structure containing the parameters about the PPP management.

```
#define PPP_AUTHSTRMAXSIZE      64
#define PPP_IPADDRSTRMAXSIZE    16

typedef struct
{
    ascii PPPMYIP[PPP_IPADDRSTRMAXSIZE+1];
    ascii PPPEERIP[PPP_IPADDRSTRMAXSIZE+1];
    ascii PPPSRVUN[PPP_AUTHSTRMAXSIZE+1];
    ascii PPPSRVPW[PPP_AUTHSTRMAXSIZE+1];
    u8     PPPMODE;
}ed_pppSetupParams_t;
```

- PPPMYIP:
  This parameter defines the IP address (32-bit number in dotted-decimal notation (i.e. xxx.xxx.xxx.xxx)) to be attributed to the TCP/IP stack when the PPP Server mode is running.

  By default this parameter is set to 0.0.0.0

  Note:
  When the TCP/IP stack behaves as a PPP server (according to the PPPMODE parameter), it is in charge of the IP address attribution mechanism.

  Once the PPP authentication successfully achieved, the remote PPP peer asks the TCP/IP stack for an IP address. The related PPP layer, called IPCP, suggests the peer an IP address previously stored in the TCP/IP stack parameters. If the remote accepts this address, the IP link is then established.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

- PPPEERIP:
  This parameter defines the IP address (32-bit number in dotted-decimal notation (i.e. xxx.xxx.xxx.xxx)) to be attributed to the remote PPP peer when the PPP Server mode is running.

  By default this parameter is set to 0.0.0.0

  Note:

  When the TCP/IP stack behaves as a PPP server (according to the PPPMODE parameter), it is in charge of the IP address attribution mechanism.

  Once the PPP authentication successfully achieved, the remote PPP peer asks the TCP/IP stack for an IP address. The related PPP layer, called IPCP, then suggests the peer an IP address previously stored in the TCP/IP stack parameters. If the remote accepts this address, the IP link is then established.

- PPPSRVUN:
  This parameter defines the login (Alpha-numeric ASCII text string up to 64 characters) that must be specified by the remote PPP client.

  By default this parameter is set to empty string

  Note:

  When the TCP/IP stack behaves as a PPP server (according to the PPPMODE parameter), it checks the remote PPP client login/password before granting access.

- PPPSRVPW:
  This parameter defines the password (alpha-numeric ASCII text string up to 64 characters) that must be specified by the remote PPP client.

  By default this parameter is set to empty string

  Note:
  When the TCP/IP stack behaves as a PPP server (according to the PPPMODE parameter), it checks the remote PPP client login/password before granting access.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

- PPPMODE:
  This parameter selects the behavior that the TCP/IP stack will have after the physical layer successful launching. These are its possible values:

Value	Description
1 (default)	(Standard PPP) the TCP/IP stack behaves as a PPP client for outgoing calls and as a PPP server for incoming calls.
2	(Reverse PPP) the TCP/IP stack behaves as a PPP server for outgoing calls and as a PPP client for incoming calls.
3	(PPP client only) the TCP/IP stack always behaves as a PPP client for both outgoing and incoming calls.
4	(PPP server only) the TCP/IP stack always behaves as a PPP server for both outgoing and incoming calls.

  By default this parameter is set to 1.

2.3.3.3 The ed_PPPSetConfig function

This function sets up some specific parameters about the PPP management.

- **Prototype**

```
s8 ed_PPPSetConfig(ed_pppSetupParams_t  *Params);
```

- **Parameters**

  Params:
  Structure containing the parameters to set up. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG
  - ED_ERR_PARAM_BAD_VALUE
  - ED_ERR_PARAM_TOO_LONG
  - ED_ERR_PARAM_WRITE_E2P
  - ED_ERR_SERVICE_ALREADY_RUNNING

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.3.3.4 The ed_PPPGetConfig function

This function gets up some specific parameters about the PPP management.

- **Prototype**

```
s8 ed_PPPGetConfig(ed_pppSetupParams_t  *Params);
```

- **Parameters**

  Params:
  Structure containing the parameters set. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG

See §3 - List of error codes and reply codes for more information on error codes.

2.3.4 DNS Management

2.3.4.1 Required Header Files

The interface header file for the DNS management service is:

ed_dns.h

2.3.4.2 The ed_dnsSetupParams_t type

Structure containing the parameters about the DNS.

```
#define DNS_IPADDRSTRMAXSIZE    15


typedef struct
{
    ascii DnsServ1[DNS_IPADDRSTRMAXSIZE+1];
    ascii DnsServ2[DNS_IPADDRSTRMAXSIZE+1];
}ed_dnsSetupParams_t;
```

- DnsServ1 and DnsServ2:
  In order to translate the server names from literal format into IP addresses, the TCP/IP stack software implements the Domain Name System (DNS) protocol. The DNS Server IP address must be specified to the TCP/IP stack software.
  It is a 32-bits number in dotted-decimal notation (i.e. xxx.xxx.xxx.xxx).

  By default this parameter is set to 0.0.0.0.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.3.4.3 The ed_DNSSetConfig function

This function sets up DNS parameters.

- **Prototype**

```
s8 ed_DNSSetConfig(ed_dnsSetupParams_t  *Params);
```

- **Parameters**

  Params:
  Structure containing the DNS parameters to set up. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG
  - ED_ERR_PARAM_BAD_VALUE
  - ED_ERR_PARAM_TOO_LONG
  - ED_ERR_PARAM_WRITE_E2P
  - ED_ERR_SERVICE_ALREADY_RUNNING

See §3 - List of error codes and reply codes for more information on error codes.

2.3.4.4 The ed_DNSGetConfig function

This function gets up DNS parameters.

- **Prototype**

```
s8 ed_DNSGetConfig(ed_dnsSetupParams_t  *Params);
```

- **Parameters**

  Params:
  Structure containing the parameters set. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.3.5 GPRS Management

2.3.5.1 Required Header Files

The interface header file for the GPRS management service is:

ed_gprs.h

2.3.5.2 The ed_gprsSetupParams_t type

Structure containing the parameters about the GPRS management.

```
#define GPRS_AUTHSTRMAXSIZE    50
#define GPRS_STRMAXSIZE        100

typedef struct
{
    u8 Cid;
    u8 Mode;
    ascii ApnServ[GPRS_STRMAXSIZE+1];
    ascii ApnUn[GPRS_AUTHSTRMAXSIZE+1];
    ascii ApnPw[GPRS_AUTHSTRMAXSIZE+1];
}ed_gprsSetupParams_t;
```

- Cid:
  This parameter defines the PDP context identifier (between 1 and 4) which specifies a particular PDP context definition.

  By default this parameter is set to 1.

  Note:
  This parameter is local and may be used in other PDP context-related commands: the ed_GPRSSetConfig() function will overwrite the corresponding global PDP context definition.

- Mode:
  This parameter configures the activation of the Wavecom software for switching between GSM or GPRS as the following:
  - 0: WAVECOM software is configured for a GSM use
  - 1: WAVECOM software is configured for a GPRS use

  By default this parameter is set to GPRS use (Mode = 1).

- ApnServ:
  This parameter defines the Access Point Name parameter (alphanumeric ASCII text string up to 100 characters) coming from the GSM operator for providing GPRS access.

  By default this parameter is set to empty string

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

- ApnUn:
  This parameter defines Access Point Name Username (Alphanumeric ASCII text string up to 50 characters) parameter coming with the ApnPw from the GSM operator for providing GPRS access.

  By default this parameter is set to empty string

- ApnPw:
  This parameter defines the Access Point Name Password parameter (Alphanumeric ASCII text string up to 50 characters) used with the ApnUn from the GSM operator for providing GPRS access.

  By default this parameter is set to empty string

2.3.5.3 The ed_GprsSetConfig function

This function sets up some specific parameters about the GPRS management.

- **Prototype**

```
s8 ed_GPRSSetConfig(ed_gprsSetupParams_t  *Params);
```

- **Parameters**

  Params:
  Structure containing the parameters to set up. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG
  - ED_ERR_PARAM_BAD_VALUE
  - ED_ERR_PARAM_TOO_LONG
  - ED_ERR_PARAM_WRITE_E2P
  - ED_ERR_MODEM_RUNNING
  - ED_ERR_SERVICE_ALREADY_RUNNING

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.3.5.4 The ed_GprsGetConfig function

This function gets up some specific parameters about the GPRS management.

- **Prototype**

```
s8 ed_GPRSGetConfig(ed_gprsSetupParams_t  *Params);
```

- **Parameters**

  Params:
  Structure containing the parameters set. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.4 SMTP/POP3 e-mail Service

IP connectivity library provides this service to handle e-mail related events, and to setup e-mail parameters.

2.4.1 SMTP e-mail services.

The SMTP service handles the sending of e-mail to a remote SMTP server.

In addition to the regular recipient (REC) field (the "To:" field), the carbon copy (CCREC) field (the "Cc:" field) is also supported. Each of these fields (REC, CCREC) may hold up to ten e-mail addresses. Each of these addresses for each field is identified by an index (between 1 and 10).

The SMTP service also offers a preset e-mail feature. It is possible to define up to three pre-recorded e-mails (by pre-setting, for each of them, the body, recipients (up to 10 e), carbon-copy (up to 10 e), the subject).

The preset e-mails are identified by their id (1, 2, 3). The id #1 e-mail settings (except the body) are also used to send non pre-recorded e-mails.

For compatibility reasons, only the first REC and CCREC address (identified by the index 1 in the list of 10 possible address) can be set with ed_EmailSetConfig(). ed_EmailAddrSet() should be used to set the 9 other REC and CCREC addresses.

2.4.1.1 Required Header Files.

The interface header file for the SMTP service is:

ed_smtp.h

2.4.1.2 The ed_EmailSetupParams_t type

Structure containing the parameters about the E-mail messages management.

```
#define SMTP_STRMAXSIZE    120


typedef struct
{
    u8     Id;
    ascii  CcRec[SMTP_STRMAXSIZE+1];
    ascii  Rec[SMTP_STRMAXSIZE+1];
    ascii  Subj[SMTP_STRMAXSIZE+1];
    ascii  Body[SMTP_STRMAXSIZE+1];
}ed_EmailSetupParams_t;
```


This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

- **Id:**
  The ID of the parameter to setup (from 1 to 3).

  NOTE: On wismo Q2406A, only ID=1 is valid.

- **CcRec:**
  This parameter contains the e-mail address (up to 120 characters) of the additional recipient with index 1. For a given email Id, up to 10 E-mail addresses may be recorded in the 'Cc:' field.
  There are two ways to access this field:
  - by using the ed_EmailSetConfig() and ed_EmailGetConfig() functions for the address with index 1 only,
  - or by using the ed_EmailAddrSet() and ed_EmailAddrGet() functions.

  By default this parameter is set to an empty string

  NOTE:
  For a given *n*, the "CCRECn" parameter is directly associated with the "RECn" parameter.
  The address must be provided in literal format (for instance dev12345678@wavecom.com).

- **Rec:**
  This parameter contains the e-mail address (text string up to 120 characters) of the recipient with index 1. For a given email Id, up to 10 e-mail addresses may be recorded in the 'To:' field. This e-mail address will appear first in the header of the e-mail sent by the TCP/IP stack software in the field ' To: '.
  There are two ways of accessing this field for read or write:
  1. by using the ed_EmailSetConfig() and ed_EmailGetConfig() functions for the address with index #1 only,
  2. or by using the ed_EmailAddrSet() and ed_EmailAddrGet() functions.

  By default this parameter is set to an empty string

  NOTE:
  The address must be provided in literal format (for instance dev12345678@wavecom.com).

- **Subj:**
  This parameter contains pre-defined one-line subject (text string of up to 120 characters) that will be used by the TCP/IP stack to compose the e-mail header.

  By default this parameter is set to empty string

- **Body:**
  These parameters store pre-defined message bodies. They allow the host application to send pre-defined e-mail combinations.

  By default this parameter is set to empty string.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.4.1.3 The ed_EmailSetConfig function

This function sets up some specific parameters about the e-mail messages management.

- **Prototype**

```
s8 ed_EmailSetConfig(ed_EmailSetupParams_t *Params);
```

- **Parameters**

  Structure containing the parameter to set up. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NO_HIGH_LEVEL_INTERNET
  - ED_ERR_NOT_INIT
  - ED_ERR_PARAM_BAD_VALUE
  - ED_ERR_PARAM_TOO_LONG
  - ED_ERR_PARAM_WRITE_E2P
  - ED_ERR_INVALID_ARG
  - ED_ERR_SERVICE_ALREADY_RUNNING

See §3 - List of error codes and reply codes for more information on error codes.

2.4.1.4 The ed_EmailGetConfig function

This function gets up some specific parameters about the e-mail messages management.

- **Prototype**

```
s8 ed_EmailGetConfig(ed_EmailSetupParams_t *Params);
```

- **Parameters**

  Structure containing the parameter to get. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NO_HIGH_LEVEL_INTERNET
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG
  - ED_ERR_PARAM_BAD_VALUE

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.4.1.5 The ed_EmailAddrParams_t type.

Structure holding an e-mail address, used to set or get one of the recipient and Carbon-copy recipient e-mail addresses (multi-recipient feature).

```
typedef struct
{
      u8       Id;
      u8       MailAddrIdx;
      ascii    MailAddr[SMTP_STRMAXSIZE+1];
} ed_EmailAddrParams_t;
```

- Id:
  The ID of the parameter to set up (from 1 to 3).

- MailAddrIdx:
  The e-mail address rank. For a given Id, 10 e-mail addresses maximum may be recorded to the 'To:' field and 10 e-mail addresses maximum may be recorded to the 'Cc:' field of the current e-mail to send.
  RANGE: [1, 10].

- MailAddr:
  The e-mail address (text string up to 120 characters) associated with a given Id and rank.
  For a given Id, by default, the recipient and the Carbon-copy recipient are empty.
  NOTE:
  Each address must be provided in literal format (for instance dev12345678@wavecom.com).

2.4.1.6 The ed_EmailAddrSet function.

For a given Id, this function sets the 'To:' or the 'Cc:' recipient according to the boolean bRecCcRecSwitch value (TRUE to set 'Cc' field, FALSE to set 'To:' field).

- **Prototype**

```
s8 ed_EmailAddrSet(ed_EmailAddrParams_t *Params, bool bRecCcRecSwitch);
```

- **Parameters**

  Params:
  Structure holding the Id, the e-mail address rank and the e-mail address (which may be an empty string). This structure is defined above.

  bRecCcRecSwitch:
  Switch defining the type of e-mail address being set. If FALSE, the e-mail address set is one out of the 10 e-mail addresses that go in the 'To:' field of the e-mail sent. If TRUE, the e-mail address set is one out of the 10 e-mail addresses that go in the 'Cc:' field of the e-mail to be sent.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NO_HIGH_LEVEL_INTERNET
  - ED_ERR_NOT_INIT
  - ED_ERR_PARAM_BAD_VALUE
  - ED_ERR_PARAM_BAD_MAILIDX
  - ED_ERR_PARAM_TOO_LONG
  - ED_ERR_PARAM_WRITE_E2P
  - ED_ERR_INVALID_ARG
  - ED_ERR_SERVICE_ALREADY_RUNNING

See §3 - List of error codes and reply codes for more information on error codes.

2.4.1.7 The ed_EmailAddrGet function.

For a given Id, this function gets the 'To:' or the 'Cc:' recipient according to the boolean bRecCcRecSwitch value (TRUE to get the 'Cc' field, FALSE to get the 'To:' field).

- **Prototype**

```
s8 ed_EmailAddrGet(ed_EmailAddrParams_t *Params, bool bRecCcRecSwitch);
```

- **Parameters**

  `Params`:
  IN: Structure holding the Id, the e-mail address rank. This structure is defined above.
  OUT: Structure holding the Id, the e-mail address rank and the e-mail address (which may be an empty string).

  `bRecCcRecSwitch:`
  Switch defining the type of the e-mail address to get. If FALSE, the e-mail address to get is one out of the 10 e-mail addresses that go in the 'To:' field of the e-mail sent. If TRUE, the e-mail address to get is one out of the 10 e-mail addresses that go in the 'Cc:' field of the e-mail to be sent.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NO_HIGH_LEVEL_INTERNET
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG
  - ED_ERR_PARAM_BAD_VALUE
  - ED_ERR_PARAM_BAD_MAILIDX

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.4.1.8 The ed_SMTPSetupParams_t type

Structure containing the parameters about the SMTP e-mail management.

```
#define SMTP_AUTHSTRMAXSIZE    64
#define SMTP_STRMAXSIZE    120


typedef struct
{
      u16      SmtpPort;
      ascii    SmtpServ[SMTP_STRMAXSIZE+1];
      ascii    Domain[SMTP_STRMAXSIZE+1];
      ascii    SmtpUn[SMTP_AUTHSTRMAXSIZE+1];
      ascii    SmtpPw[SMTP_AUTHSTRMAXSIZE+1];
      ascii    SenderName[SMTP_STRMAXSIZE+1];
      ascii    SenderAddr[SMTP_STRMAXSIZE+1];
}ed_SMTPSetupParams_t;
```

- SmtpPort:
  This parameter contains the port of the SMTP server used for the e-mail sending. **It should only be changed upon request of your network administrator** as it applies to network infrastructure including firewalls, proxy or specific TCP port translation.

  SYNTAX:

  From 1 to 5 numeric digits (0 to 9).
  By default, it is set to 25.

  Notes:
  - numbers above 65,535 are illegal as the port identification fields are 16 bits long in the TCP header.
  - port number 0 is illegal.

- SmtpServ:
  This parameter contains the address of the SMTP server that is to be used to send e-mail messages. In most cases, the local ISP maintains the SMTP server.

  SYNTAX:
  32-bit number in dotted-decimal notation (i.e. xxx.xxx.xxx.xxx) or alphanumeric ASCII text string up to 120 characters if DNS is available.

  By default it is set to empty string

- Domain:
  When sending an e-mail message, the TCP/IP stack software must provide the SMTP server with the domain name of the sender. In some cases, this domain name may be different from the domain name included in the sender's e-mail address.
  This parameter contains the different domain name (text string up to 120 characters).

  By default it is set to empty string

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

- **SmtpUn:**
  To send e-mail messages, some SMTP servers use an authentication process. In these cases, the TCP/IP stack software will provide the SMTP user name (associated with a SMTP password) for the e-mail sending process. This parameter holds the SMTP user name (ASCII text string up to 64 characters). If this parameter is an empty string, the authentication mode is inactive. If this parameter and the SmtpPw parameter are both not empty, the authentication mode is active.

  Note: the IP Connectivity library only supports the LOGIN authentication mechanism.

  By default this parameter is set to empty string

- **SmtpPw:**
  This parameter holds the SMTP password (ASCII text string up to 64 characters) associated to **SmtpUn** parameter (see above description).

  By default this parameter is set to empty string

- **SenderName:**
  This parameter appears in the field: 'From: ' of the header of the e-mail sent by the TCP/IP stack software. It is the sender's literal name (up to 120 characters). It is different from the **SenderAddr** parameter, which is the sender's e-mail address.

  By default this parameter is set to empty string

  Note:
  The address must be provided in literal format (for instance "machine 245").

- **SenderAddr:**
  This parameter contains the e-mail identification of the hardware platform itself or the optional attached equipment.

  By default this parameter is set to empty string

  Note:
  This e-mail address will appear in the header of the e-mail sent by the TCP/IP stack software, in the field "From: "

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.4.1.9 The ed_SMTPSetConfig function

This function sets up some specific parameters about the SMTP e-mail management.

- **Prototype**

```
s8 ed_SMTPSetConfig(ed_SMTPSetupParams_t *Params);
```

- **Parameters**

  Structure containing the parameter to set up. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NO_HIGH_LEVEL_INTERNET
  - ED_ERR_NOT_INIT
  - ED_ERR_PARAM_BAD_VALUE
  - ED_ERR_PARAM_TOO_LONG
  - ED_ERR_PARAM_WRITE_E2P
  - ED_ERR_INVALID_ARG
  - ED_ERR_SERVICE_ALREADY_RUNNING

  See §3 - List of error codes and reply codes for more information on error codes.

2.4.1.10 The ed_SMTPGetConfig function

This function gets up some specific parameters about the SMTP e-mail management.

- **Prototype**

```
s8 ed_SMTPGetConfig(ed_SMTPSetupParams_t *Params);
```

- **Parameters**

  Structure containing the parameter to get. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NO_HIGH_LEVEL_INTERNET
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG

  See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.4.1.11 The ed_SendMail function

This function sends one of the 3 pre-defined e-mail messages to the SMTP server.

Once an IP link is established, the attached host can direct the TCP/IP stack to send an e-mail message at any time (except when the TCP/IP stack software is already in a process using TCP resources).

The header of this e-mail is built using the **REC1/2/3**, **CCREC1/2/3** and **SUBJ1/2/3** parameters while the body is filled in the **BODY1/2/3** parameter.

This command is similar to a "send e-mail" operation issued by a standard messaging client on a PC.

- **Prototype**

```
s8 ed_SendMail (u8 Id, ed_ResponseCbk_f ResponseCbk);
```

- **Parameters**

  Id:

  The number of the pre-defined e-mail to send (from 1 to 3).

  ResponseCbk:

  ResponseCbk is called by the IP Connectivity library after the e-mail has been successfully sent or when an error occurred.

  This callback function is defined as below:

```
typedef void (* ed_ ResponseCbk_f) (s32 ResponseCode,
                                    TeDHandle id);
```

  TeDHandle id: always set to ED_ID_SMTP_SENDMAIL.

  ResponseCode values:

  - ED_OK_MAIL_SENT:
  - ED_ERR_DISTANT_DNS:
  - ED_ERR_STACK_INTERNAL:
  - ED_ERR_DISTANT_NO_RESP:
  - ED_ERR_DISTANT_OPEN:
  - ED_ERR_DISTANT_CCREC_ADDR:
  - ED_ERR_DISTANT_REC_ADDR:
  - ED_ERR_DISTANT_SENDER_ADDR
  - ED_ERR_DISTANT_BODY:
  - ED_ERR_DISTANT_SEND:
  - ED_ERR_DISTANT_CLOSE:
  - ED_ERR_DISTANT_TCP_CLOSED:
  - ED_ERR_DISTANT_TCP_CLOSED_BYPEER:
  - ED_ERR_DISTANT_USERNAME:
  - ED_ERR_DISTANT_PASSWORD:

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

- ED_ERR_NETWORK_KO:

  GSM network is lost, SMTP e-mail sending is aborted

- ED_ERR_GPRS_SESSION_LOST:

  GPRS session is lost, SMTP e-mail sending is aborted.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:

  - ED_ERR_NO_HIGH_LEVEL_INTERNET
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG
  - ED_ERR_STACK_BAD_CONFIG:
  - ED_ERR_PHY_NOT_ACTIVATED:
  - ED_ERR_STACK_BUSY:

  See §3 - List of error codes and reply codes for more information on error codes.

2.4.1.12 The ed_PutMail function

This function allows the attached application to send an e-mail message containing body text passed to the TCP/IP stack directly with the ed_SendData() function (c.f. section 2.9.1).

Once an IP link is established, the attached host can send an e-mail message at any time (if the TCP/IP stack software is not already in a process using TCP resources).

The header of this e-mail is built using the **Rec1** (see § 2.4.1.6), **CcRec1** (see § 2.4.1.6) and **Subj1** (see § 2.4.1.3) parameters.

This function is similar to a "send e-mail" operation issued by a standard messaging client on a PC.

- **Prototype**

```
s8 ed_PutMail (ed_ResponseCbk_f ResponseCbk, ed_DataRequest_f DataRequest);
```

- **Parameters**

  ResponseCbk:

  ResponseCbk is called by the IP Connectivity library when the SMTP session is opened or when an error occurs.

  This callback function is defined as below:

```
typedef void (* ed_ ResponseCbk_f) (s32 ResponseCode,
                                    TeDHandle id);
```

  TeDHandle id: always set to ED_ID_SMTP_PUTMAIL.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

ResponseCode values:

- ED_OK_MAIL_SENT

  The mail has been successfully sent. The PutMail session is over.

- ED_INFO_WAITING_FOR_DATA

  The TCP/IP stack is ready to send data to SMTP server. TCP/IP stack immediately transfers all the data to the SMTP server (using the DataRequest callback).

- ED_ERR_DISTANT_DNS

  The address of the SMTP server has not been resolved by the secondary DNS server. TCP/IP stack is not able to reach the primary and secondary DNS servers or a wrong SMTP server address has been filled in.

Possible error codes are:

- ED_ERR_STACK_INTERNAL
- ED_ERR_DISTANT_NO_RESP
- ED_ERR_DISTANT_OPEN
- ED_ERR_DISTANT_CCREC_ADD
- ED_ERR_DISTANT_REC_ADDR
- ED_ERR_DISTANT_SENDER_ADDR
- ED_ERR_DISTANT_BODY
- ED_ERR_DISTANT_SEND
- ED_ERR_DISTANT_CLOSE
- ED_ERR_DISTANT_TCP_CLOSED
- ED_ERR_DISTANT_TCP_CLOSED_BYPEER
- ED_ERR_DISTANT_USERNAME
- ED_ERR_DISTANT_PASSWORD
- ED_ERR_NETWORK_KO
- ED_ERR_GPRS_SESSION_LOST

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

DataRequest:

DataRequest is called by the IP Connectivity library to inform the client that it can send new data. It is called as soon as the SMTP session is opened, to start the data transfer.

This callback function is defined as follows:

```
typedef void (* ed_DataRequest_f) (u16 MaxLen);
```

where **MaxLen** is the maximum number of bytes that can be sent (using the ed_SendData() function (c.f. section 2.9.1 below)).

Note: it is not necessary to call ed_SendData() as soon as the DataRequest callback is triggered. The ed_SendData() call may occur later.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NO_HIGH_LEVEL_INTERNET
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG
  - ED_ERR_STACK_BAD_CONFIG
  - ED_ERR_PHY_NOT_ACTIVATED
  - ED_ERR_STACK_BUSY

  See §3 - List of error codes and reply codes for more information on error codes.

2.4.2 POP3 service.

The POP3 service handles the retrieving of e-mail from a remote POP3 server.

2.4.2.1 Required Header Files.

The interface header file for the POP3 service is:

ed_pop3.h

2.4.2.2 The ed_POP3SetupParams_t type

Structure containing the parameters about the POP3 e-mail management.

```
#define POP3_AUTHSTRMAXSIZE     64
#define POP3_STRMAXSIZE         120
```


This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

```
typedef struct
{
      u16      Pop3Port;
      bool     bPop3HeaderMode;
      ascii    Pop3Serv[POP3_STRMAXSIZE+1];
      ascii    Pop3Un[POP3_AUTHSTRMAXSIZE+1];
      ascii    Pop3Pw[POP3_AUTHSTRMAXSIZE+1];
}ed_POP3SetupParams_t;
```

- bPop3HeaderMode:
  This parameter configures the TCP/IP stack to send or not the POP3 header to the application. The POP3 header contains the From, Cc and Subject fields.
  - 0: the e-mail header will not be sent to the application while retrieving
  - 1: the e-mail header will be sent to the application while retrieving

  By default, this parameter is set to 1.

- Pop3Port:
  This parameter contains the POP3 server port used for e-mail retrieving.

  By default, this parameter is set to 110.

  Syntax:

  5 numeric digits. Note that numbers above 65 535 are illegal as the port identification fields are 16 bits long in the TCP header. Note that port number 0 is illegal too.

  Note:
  This parameter should be changed only upon request of your network administrator. It applies to network infrastructures including Firewalls, Proxy or specific TCP port translation.

- Pop3Serv:
  This parameter contains the address of the POP3 server that is to be used. The POP3 server must be the one where the specified e-mail account is hosted (which is not necessarily maintained by the local ISP).

  By default this parameter is set to empty string

  Syntax:
  32-bit number in dotted-decimal notation (i.e. xxx.xxx.xxx.xxx) or alphanumeric ASCII text string up to 120 characters if DNS is available.

- Pop3Pw:
  This parameter contains the password (ASCII text string up to 64 characters) for POP3 account. To retrieve e-mail messages sent to a specified e-mail address, the TCP/IP stack software must know the POP3 password that has been set for that e-mail account.

  By default this parameter is set to empty string

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

- Pop3Un:
  This parameter contains the user name (ASCII text string up to 64 characters) for POP3 account. To retrieve e-mail messages sent to a specified e-mail address, the TCP/IP stack software must know the POP3 user name that has been set for that e-mail account.

  By default this parameter is set to empty string

2.4.2.3 The ed_POP3SetConfig function

This function sets up some specific parameters about the POP3 e-mail management.

- **Prototype**

```
s8 ed_POP3SetConfig(ed_POP3SetupParams_t *Params);
```

- **Parameters**

Structure containing the parameter to set up. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NO_HIGH_LEVEL_INTERNET
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG
  - ED_ERR_PARAM_BAD_VALUE
  - ED_ERR_PARAM_TOO_LONG
  - ED_ERR_PARAM_WRITE_E2P
  - ED_ERR_SERVICE_ALREADY_RUNNING

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.4.2.4 The ed_POP3GetConfig function

This function gets up some specific parameters about the POP3 e-mail management.

- **Prototype**

```
s8 ed_POP3GetConfig(ed_POP3SetupParams_t *Params);
```

- **Parameters**

  Structure containing the parameter to get. This structure is defined above.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NO_HIGH_LEVEL_INTERNET
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG

See §3 - List of error codes and reply codes for more information on error codes.

2.4.2.5 The ed_GetMail function

This function allows the IP connectivity library user to initiate the retrieval of the first mail on the POP3 account specified in the POP3 settings.

Once an IP link is established, the IP connectivity library user can retrieve an e-mail message at any time (except when the TCP/IP stack software is already in a process using TCP resources).

This function is similar to a "check e-mail box" feature issued by a standard messaging client on a PC.

Note that the e-mail is deleted from the POP3 server after retrieval.

- **Prototype**

```
s8 ed_GetMail (ed_ResponseCbk_f ResponseCbk,ed_DataHandler_f DataHnd);
```

- **Parameters**

  ResponseCbk:

  ResponseCbk is called by the IP connectivity library when the POP3 connection is opened or when an error occurred.

  This callback function is defined as below:

```
typedef void (* ed_ ResponseCbk_f) (s32 ResponseCode,
                                    TeDHandle id);
```

  TeDHandle id: always set to ED_ID_POP3_GETMAIL.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

ResponseCode values:

- ED_INFO_MAIL

  This response is issued when at least one e-mail message is located in the specified POP3 mailbox. It is also an indication that the e-mail reception begins. Depending on the bPop3HeaderMode parameter(see §2.4.2.2), the TCP/IP stack sends the e-mail header to the data handler DataHnd.

- ED_INFO_NOMAIL
- ED_OK_POP3
- ED_ERR_DISTANT_DNS
- ED_ERR_STACK_INTERNAL
- ED_ERR_DISTANT_USERNAME
- ED_ERR_DISTANT_PASSWORD
- ED_ERR_DISTANT_NO_RESP
- ED_ERR_DISTANT_OPEN
- ED_ERR_DISTANT_CLOSE
- ED_ERR_DISTANT_TCP_CLOSED
- ED_ERR_DISTANT_TCP_CLOSED_BYPEER
- ED_ERR_DISTANT_DATA_RETR
- ED_ERR_DISTANT_EMAIL_RETR
- ED_ERR_DISTANT_EMAIL_HDR_RETR
- ED_ERR_DISTANT_DEL
- ED_ERR_NETWORK_KO
- ED_ERR_GPRS_SESSION_LOST

DataHnd:

DataHnd is called by the IP connectivity library to send the POP3 received data.

At the end of the e-mail retrieval, IP connectivity library calls pfDataHnd with Data pointer set to NULL and DataLen set to 0.

This callback function is defined as:

```
typedef u16 (* ed_DataHandler_f ) (u16 DataLen, u8 * Data,
                                   TeDHandle id);
```

where:

- ::IN: u8 * Data: Data pointer; freed by eDlib.
- IN: u16 DataLen: number of available Bytes in u8 *Data.
- IN: TeDHandle id: always set to ED_ID_POP3_GETMAIL.
- RETURN: Number of bytes “consumed” by the callback. The data left will be handed over at the next call, with possible new data.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - ED_ERR_NO_HIGH_LEVEL_INTERNET
  - ED_ERR_NOT_INIT
  - ED_ERR_INVALID_ARG
  - ED_ERR_STACK_BAD_CONFIG
  - ED_ERR_PHY_NOT_ACTIVATED
  - ED_ERR_STACK_BUSY

  See §3 - List of error codes and reply codes for more information on error codes.

2.5 FTP Service

IP connectivity library provides this service to handle FTP related events and setup FTP parameters.

2.5.1 Required Header Files.

The interface header file for the FTP service is:

ed_ftp.h

2.5.2 The ed_FTPSetupParams_t type

Structure containing the parameters about the FTP management.

```
#define FTP_AUTHSTRMAXSIZE      64
#define FTP_STRMAXSIZE          120

typedef struct
{
      u16       FtpPort;
      u8        FtpMode;
      ascii     FtpType;
      ascii     FtpServ[FTP_STRMAXSIZE+1];
      ascii     FtpUn[FTP_AUTHSTRMAXSIZE+1];
      ascii     FtpPw[FTP_AUTHSTRMAXSIZE+1];
}ed_FTPSetupParams_t;
```


This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

- **FtpPort:**
  This parameter contains the control port of the FTP server used for file transfer.

  By default, this parameter is set to 21.

  Syntax:

  numeric 1 to 5 digits. Note that numbers above 65535 are illegal as the port identification fields are 16 bits long in the TCP header. Note that 0 is illegal too.

  Note:
  This parameter should be changed only upon request of your network administrator. It applies for network infrastructure including Firewalls, Proxy or specific TCP port translation.
- **FtpMode:**
  This parameter contains the FTP behavior.

  By default, this parameter is set to 0.

  Syntax:

  - 0: for FTP active mode
  - 1: for FTP passive mode

- **FtpType:**
  This parameter contains the type of data to be transferred within the FTP session.

  Syntax:
  - A: for FTP ASCII sessions
  - I: for FTP Binary sessions

  By default, this parameter is set to I.

  Note:
  This parameter must be set before transferring files from a specified FTP server.

- **FtpServ:**
  This parameter contains the address of the FTP server that is to be used to download files.

  Syntax:
  32-bit number in dotted-decimal notation (i.e. xxx.xxx.xxx.xxx) or alphanumeric ASCII text string up to 120 characters if DNS is available.

  By default this parameter is set to empty string

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

- FtpUn:
  This parameter contain the FTP user name of the FTP session (ASCII text string up to 64 characters).

  By default this parameter is set to empty string

- FtpPw:
  This parameter contain the FTP password of the FTP session (ASCII text string up to 64 characters).

  By default this parameter is set to empty string

wavecom confidential ©
This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.5.3 The ed_FTPSetConfig function

This function sets up some specific parameters about the FTP management.

2.5.3.1 Prototype

```
s8 ed_FTPSetConfig(ed_FTPSetupParams_t *Params);
```

2.5.3.2 Parameters

Structure containing the parameter to set up. This structure is defined above.

2.5.3.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NO_HIGH_LEVEL_INTERNET
- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG
- ED_ERR_PARAM_BAD_VALUE
- ED_ERR_PARAM_TOO_LONG
- ED_ERR_PARAM_WRITE_E2P
- ED_ERR_SERVICE_ALREADY_RUNNING

See §3 - List of error codes and reply codes for more information on error codes.

2.5.4 The ed_FTPGetConfig function

This function gets up some specific parameters about the FTP management.

2.5.4.1 Prototype

```
s8 ed_FTPGetConfig(ed_FTPSetupParams_t *Params);
```

2.5.4.2 Parameters

Structure containing the parameter to get. This structure is defined above.

2.5.4.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NO_HIGH_LEVEL_INTERNET
- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.5.5 The ed_FTPGetFileParams_t type

Structure containing the parameters about the FTP management to get a file from an FTP server.

```
typedef struct
{
      ascii    FtpGetFilename[FTP_STRMAXSIZE+1];
      ascii    FtpGetPath[FTP_STRMAXSIZE+1];
}ed_FTPGetFileParams_t;
```

- FtpGetFilename:
  This parameter contains the name (ASCII text string up to 120 characters) of the file to download from the FTP server.

  By default this parameter is set to empty string

- FtpGetPath:
  This parameter contains the path (ASCII text string up to 120 characters) of the relevant file to download from the FTP server. For exemple it could be "/wavecom".

  By default this parameter is set to "." (for local directory).

  Note:
  Depending on the FTP server, the value . can be used for getting a file from the root directory of the FTP server

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.5.6 The ed_FTPGetFileSetConfig function

This function sets up some specific parameters about the FTP download.

2.5.6.1 Prototype

```
s8 ed_FTPGetFileSetConfig(ed_FTPGetFileParams_t *Params);
```

2.5.6.2 Parameters

Structure containing the parameter to set up. This structure is defined above.

2.5.6.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NO_HIGH_LEVEL_INTERNET
- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG
- ED_ERR_PARAM_BAD_VALUE
- ED_ERR_PARAM_TOO_LONG
- ED_ERR_PARAM_WRITE_E2P
- ED_ERR_SERVICE_ALREADY_RUNNING

See §3 - List of error codes and reply codes for more information on error codes.

2.5.7 The ed_FTPGetFileGetConfig function

This function gets up some specific parameters about the FTP download.

2.5.7.1 Prototype

```
s8 ed_FTPGetFileGetConfig(ed_FTPGetFileParams_t *Params);
```

2.5.7.2 Parameters

Structure containing the parameter to get. This structure is defined above.

2.5.7.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NO_HIGH_LEVEL_INTERNET
- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.5.8 The ed_FTPPutFileParams_t type

Structure containing the parameters about the FTP management to put a file on an FTP server.

```
typedef struct
{
      ascii    FtpPutFilename[FTP_STRMAXSIZE+1];
      ascii    FtpPutPath[FTP_STRMAXSIZE+1];
}ed_FTPPutFileParams_t;
```

- FtpPutFilename:
  This parameter contains the name (ASCII text string up to 120 characters) of the file to upload to the FTP server.

  By default this parameter is set to empty string

- FtpPutPath:
  This parameter contains the path (ASCII text string up to 120 characters) of the relevant file to upload to the FTP server. For exemple it could be "/wavecom".

  By default this parameter is set to "."

  Note:
  Depending on the FTP server, the value . can be used for getting a file from the root directory of the FTP server

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.5.9 The ed_FTPPutFileSetConfig function

This function sets up some specific parameters about the FTP upload.

2.5.9.1 Prototype

```
s8 ed_FTPPutFileSetConfig(ed_FTPPutFileParams_t *Params);
```

2.5.9.2 Parameters

Structure containing the parameter to set up. This structure is defined above.

2.5.9.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NO_HIGH_LEVEL_INTERNET
- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG
- ED_ERR_PARAM_BAD_VALUE
- ED_ERR_PARAM_TOO_LONG
- ED_ERR_PARAM_WRITE_E2P
- ED_ERR_SERVICE_ALREADY_RUNNING

See §3 - List of error codes and reply codes for more information on error codes.

2.5.10 The ed_FTPPutFileGetConfig function

This function gets up some specific parameters about the FTP upload.

2.5.10.1 Prototype

```
s8 ed_FTPPutFileGetConfig(ed_FTPPutFileParams_t *Params);
```

2.5.10.2 Parameters

Structure containing the parameter to get. This structure is defined above.

2.5.10.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NO_HIGH_LEVEL_INTERNET
- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.5.11 The ed_FTPGet function

This function allows the TCP/IP stack to connect to the specified FTP server and to retrieve the specified file from this server. Once the operation has completed, the TCP/IP stack closes the FTP connection.

Once an IP link established, the attached application can retrieve a file from a FTP server at any time (except when the TCP/IP stack software is already in a process using TCP resources).

This command is similar to a GET operation (with an automatic connect/disconnect) issued by a standard FTP client on a PC. The TCP/IP stack handles the global FTP get process by itself

2.5.11.1 Prototype

```
s8 ed_FTPGet (ed_ResponseCbk_f ResponseCbk,ed_DataHandler_f DataHnd);
```

2.5.11.2 Parameters

ResponseCbk:

ResponseCbk is called by the IP connectivity library once the FTP connection is opened or once the file is fully retrieved or when an error occurs.

This callback function is defined as below:

```
typedef void (* ed_ ResponseCbk_f) (s32 ResponseCode,
                                    TeDHandle id);
```

TeDHandle id: always set to ED_ID_FTP_GET.

ResponseCode values:

- ED_OK_FILE_TRANSFERED
- ED_INFO_DATA_BEGIN
- ED_ERR_DISTANT_DNS
- ED_ERR_STACK_INTERNAL
- ED_ERR_DISTANT_USERNAME
- ED_ERR_DISTANT_PASSWORD
- ED_ERR_DISTANT_NO_RESP
- ED_ERR_DISTANT_OPEN
- ED_ERR_DISTANT_CD
- ED_ERR_DISTANT_CLOSE
- ED_ERR_DISTANT_TCP_CLOSED
- ED_ERR_DISTANT_TCP_CLOSED_BYPEER
- ED_ERR_DISTANT_DATA_RETR
- ED_ERR_NETWORK_KO
- ED_ERR_GPRS_SESSION_LOST

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

DataHnd:

DataHnd is called by the IP connectivity library to send the FTP retrieve file.

At the end of FTP received datas, IP connectivity library calls pfDataHnd with Data pointer set to NULL and DataLen set to 0.

This callback function is defined as below:

```
typedef u16 (* ed_DataHandler_f ) (u16 DataLen, u8 * Data,
                                   TeDHandle id);
```

where:

- ::IN: u8 * Data: Data pointer; freed by eDlib.
- IN: u16 DataLen: number of available Bytes in u8 *Data.
- IN: TeDHandle id: always set to ED_ID_FTP_GET.
- RETURN: Number of bytes "consumed" by the callback. The data left will be handed over at the next call, maybe with new data.

2.5.11.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NO_HIGH_LEVEL_INTERNET
- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG
- ED_ERR_STACK_BAD_CONFIG
- ED_ERR_PHY_NOT_ACTIVATED
- ED_ERR_STACK_BUSY

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.5.12 The ed_FTPPut function

This function allows the TCP/IP stack to connect to the specified FTP server and to upload the data using the ed_SendData() function (see §2.9.1) to the specified file on this server. Once the operation completed, the TCP/IP stack closes the FTP connection.

Once an IP link is established, the attached application can send a file to a FTP server at any time (except when the TCP/IP stack software is already in a process using TCP resources).

This command is similar to a PUT operation (with an automatic connect/disconnect) issued by a standard FTP client on a PC. The TCP/IP stack handles the global FTP put process by itself

2.5.13 Prototype

```
s8 ed_FTPPut (ed_ResponseCbk_f ResponseCbk, ed_DataRequest_f DataRequest);
```

2.5.13.1 Parameters

ResponseCbk:

ResponseCbk is called by the IP connectivity library when the FTP session is opened or once the file is fully sent or when an error occurs.

This callback function is defined as below:

```
typedef void (* ed_ResponseCbk_f) (s32 ResponseCode,
                                   TeDHandle id);
```

TeDHandle id: always set to ED_ID_FTP_PUT.

ResponseCode values:

- ED_OK_FILE_TRANSFERED
- ED_INFO_WAITING_FOR_DATA
- ED_ERR_DISTANT_DNS
- ED_ERR_STACK_INTERNAL
- ED_ERR_DISTANT_NO_RESP
- ED_ERR_DISTANT_OPEN
- ED_ERR_DISTANT_CD
- ED_ERR_DISTANT_SEND
- ED_ERR_DISTANT_CLOSE
- ED_ERR_DISTANT_TCP_CLOSED
- ED_ERR_DISTANT_TCP_CLOSED_BYPEER
- ED_ERR_DISTANT_USERNAME
- ED_ERR_DISTANT_PASSWORD
- ED_ERR_NETWORK_KO
- ED_ERR_GPRS_SESSION_LOST

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

DataRequest:

DataRequest is called by the IP connectivity library to inform the client that it can send new data.

This function is called when TCP session is opened to start data transfer.

This callback function is defined as below:

```
typedef void (* ed_DataRequest_f ) (u16 MaxLen, TeDHandle id);
```

where

TeDHandle id: always set to ED_ID_FTP_PUT.

MaxLen is the maximum number of bytes that can be sent now (using the ed_SendData() function).

2.5.13.2 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NO_HIGH_LEVEL_INTERNET
- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG
- ED_ERR_STACK_BAD_CONFIG
- ED_ERR_PHY_NOT_ACTIVATED
- ED_ERR_STACK_BUSY

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.6 TCP SOCKET Service

IP connectivity library provides this service to handle the TCP Sockets related events, and to setup the TCP Sockets parameters.

2.6.1 Required Header Files.

The interface header file for the TCP SOCKET service is:

ed_socket.h

2.6.2 The ed_SocketSetupParams_t type

Structure containing the parameters about the TCP Socket management.

#define SOCK_STRMAXSIZE 120

```
typedef struct
{
      u16       TcpPort;
      u16       TcpTxDelay;
      ascii     TcpServ[SOCK_STRMAXSIZE+1];
}ed_SocketSetupParams_t;
```

- TcpPort:
  This parameter holds:
  - the remote peer TCP port used for a "client" TCP session(i.e. the IP connectivity library opens a remote listening port)

  OR

  - the local listening TCP port for a "server" TCP session(i.e. the IP connectivity library opens a local port in listen mode and waits for a remote peer to connect).

  By default, this parameter is set to 0.

  Syntax:

  1 to 5 numeric digits. Note that numbers above 65 535 are illegal as the port identification fields are 16 bits long in the TCP header. Note that port number 0 is illegal too.

- TcpTxDelay:
  This parameter holds the delay (expressed in milliseconds) before the socket flushes the characters to transmit to the remote entity.

  Explanation: the IP connectivity library user provides data to transmit to the remote by using the ed_SendDataExt() function (see the pfDataRequest field of the ed_SocketTCPStart() function for more information on the full mechanism). Suppose the IP connectivity library user calls ed_SendDataExt() and provides to it one byte at every call. Internally, the IP connectivity library will not immediately send the byte (it would congest the internet network): it will "wait" until it has a reasonable amount of bytes to transmit. However, if the IP connectivity library user did not give data for TcpTxDelay ms., the IP

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

connectivity library will transmit these bytes, even if the number of bytes is low.

By default, this parameter is set to 100 ms.

Syntax:

[0, 32760] in 20 ms steps.

- TcpServ:
For a “client” TCP session (i.e. the IP connectivity library opens a remote listening port), this parameter contains the address of the remote TCP peer that is to be used. The “255.255.255.255” special IP address is forbidden.
For a “server” TCP session(i.e. the IP connectivity library opens a local port in listen mode and waits for a remote peer to connect), this parameter is used to apply filtering of incoming TCP requests from a remote destination IP address. Only requests from the configured IP address will be allowed to connect to the Wavecom module. In case of no Filtering the TcpServ addr must be set to the "255.255.255.255" value.
“0.0.0.0” is an invalid IP address for both “client” and “server” configuration.

  Syntax:
  32-bit number in dotted-decimal notation (i.e. xxx.xxx.xxx.xxx) or alphanumeric ASCII text string up to SOCK_STRMAXSIZE characters if DNS is integrated.

  By default, this parameter is set to empty string.

2.6.3 Socket Id type

Socket identifier used in all socket TCP API is one of

{ED_ID_TCPSOCKET_1, ED_ID_TCPSOCKET_2,
ED_ID_TCPSOCKET_3, ED_ID_TCPSOCKET_4,
ED_ID_TCPSOCKET_5, ED_ID_TCPSOCKET_6,
ED_ID_TCPSOCKET_7, ED_ID_TCPSOCKET_8,
ED_ID_TCPSOCKET_9, ED_ID_TCPSOCKET_10}

Notes : On Q2406A modules, only ED_ID_TCPSOCKET_1 and ED_ID_TCPSOCKET_2 is allowed.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.6.4 The ed_SocketSetConfig function

This function sets up some specific parameters about the TCP Socket management.

2.6.4.1 Prototype

```
s8 ed_SocketSetConfig(TeDHandle id, ed_SocketSetupParams_t *Params);
```

2.6.4.2 Parameters

Id:
Socket id value (Cf 2.6.3).

Params:
Structure containing the parameter to set up. This structure is defined above.

2.6.4.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG
- ED_ERR_PARAM_BAD_VALUE
- ED_ERR_PARAM_TOO_LONG
- ED_ERR_PARAM_WRITE_E2P
- ED_ERR_INVALID_ARG

  in case of argument is invalid (NULL pointer)
- ED_ERR_SERVICE_ALREADY_RUNNING

  the input parameter cannot be set because this SocketTCP is running.
- ED_ERR_INVALID_ID

  the id parameter is invalid.

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.6.5 The ed_SocketGetConfig function

This function gets up some specific parameters about the Socket management.

2.6.5.1 Prototype

```
s8 ed_SocketGetConfig(TeDHandle id, ed_SocketSetupParams_t *Params);
```

2.6.5.2 Parameters

Id:
Socket id value (Cf 2.6.3).

Params:
Structure containing the parameter to get. This structure is defined above.

2.6.5.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG
- ED_ERR_INVALID_ID

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.6.6 The ed_SocketTCPStart function

This function is used to open a remote TCP socket or to start a listening TCP socket, if the device is connected.

Once an IP link is established, the attached application can open a listening TCP socket at any time (except when the TCP/IP stack software is already in a process using TCP resources).

If this function was used to start a listening TCP socket (listen==1), the TCP connection will be active upon reception of a TCP connection request sent by a remote allowed TCP peer (TcpServ) on the appropriate TCP port (TcpPort).

2.6.6.1 Prototype

```
s8 ed_SocketTCPStart (TeDHandle id,
                      u8 Listen
                      ed_ResponseCbk_f pfResponseCbk,
                      ed_DataHandler_f pfDataHnd,
                      ed_DataRequest_f pfDataRequest);
```

2.6.6.2 Parameters

Id:
Socket id value (Cf 2.6.3).

Listen:

set to 1 if a listening TCP socket is to be started

Else IP connectivity library tries to open a TCP connection to the specified TCP server.

pfResponseCbk:

pfResponseCbk is called by the IP connectivity library when the Socket TCP connection is opened or when an error occurred.

This callback function is defined as below:

typedef void (* ed_ ResponseCbk_f) (s32 ResponseCode, TeDHandle id**);**

TeDHandle id : Socket id value (Cf 2.6.3).

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

ResponseCode values:

- ED_INFO_LISTEN_SET
- ED_OK_TCP_CLOSED
- ED_INFO_WAITING_FOR_DATA
- ED_ERR_DISTANT_DNS
- ED_ERR_DISTANT_OPEN
- ED_ERR_STACK_INTERNAL
- ED_ERR_LISTEN_STOP
- ED_ERR_DISTANT_NO_RESP
- ED_ERR_DISTANT_SEND
- ED_ERR_DISTANT_CLOSE
- ED_ERR_DISTANT_TCP_CLOSED
- ED_ERR_DISTANT_TCP_CLOSED_BY_PEER
- ED_ERR_NETWORK_KO
- ED_ERR_GPRS_SESSION_LOST

pfDataHnd:

pfDataHnd is called by the IP connectivity library to send data received from the remote to the application.

When the socket is closed, IP connectivity library calls pfDataHnd with Data pointer set to NULL and DataLen set to 0.

This callback function is defined as below:

```
typedef u16 (* ed_DataHandler_f ) (u16 DataLen, u8 * Data,
                                   TeDHandle id);
```

where:

- ::IN: u8 * Data: Data pointer; freed by eDlib.
- IN: u16 DataLen: number of available Bytes in u8 *Data.
- IN: TeDHandle id value (Cf 2.6.3).
- RETURN: Number of bytes “consumed” by the callback. The data left will be handed over at the next call, maybe with new data.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

pfDataRequest:

pfDataRequest is called by the IP connectivity library to inform the application that it can send new data.

This function is called by the IP connectivity library as soon as the TCP session is opened to initiate the data transfer.

This callback function is defined as below:

```
typedef void (* ed_DataRequest_f) (u16 MaxLen, TeDHandle id);
```

where

TeDHandle id value (Cf 2.6.3).

MaxLen is the maximum number of bytes that can be sent now (using the ed_SendData() function).

2.6.6.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG
- ED_ERR_STACK_BAD_CONFIG
- ED_ERR_PHY_NOT_ACTIVATED
- ED_ERR_STACK_BUSY
- ED_ERR_INVALID_ID

  the id parameter is invalid (whether its value is wrong or the corresponding socket is already active)
- ED_ERR_MEMALLOC_FAILED

  Not enough memory left to run a new socket.
- ED_ERR_SOCKET_CREATION_BUSY

  There can only be one socket creation at a time.

See §3 - List of error codes and reply codes for more information on error codes.

2.6.7 The ed_SocketLTCPStop function

This function allows to close a TCP socket in listening mode. It should have been previously launched by the ed_SocketTCPStart command with listen parameter set to 1).

2.6.7.1 Prototype

```
s8 ed_SocketLTCPStop (TeDHandle id, ed_ResponseCbk_f pfResponseCbk);
```


This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.6.7.2 Parameters

Id:
Socket id value (Cf 2.6.3).

pfResponseCbk:

pfResponseCbk is called by the IP connectivity library when the Socket TCP connection is closed or when an error occurred.

This callback function is defined as below:

```
typedef void (* ed_ ResponseCbk_f) (s32 ResponseCode,
                                    TeDHandle id);
```

TeDHandle id value (Cf 2.6.3).

ResponseCode value:

- ED_OK_TCP_CLOSED

2.6.7.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG
- ED_ERR_NO_LISTEN
- ED_ERR_INVALID_ID

  the id parameter is invalid

See §3 - List of error codes and reply codes for more information on error codes.

2.7 UDP SOCKET Service

The IP Connectivity library provides this service to handle UDP Socket related events, and to setup UDP Socket parameters.

2.7.1 Required Header Files.

The interface header file for the UDP SOCKET service is:

ed_udpsocket.h

2.7.2 The ed_UdpSocketSetupParams_t type

Structure containing the parameters about the UDP Socket management.

```
#define UDPSOCK_STRMAXSIZE      120
```


This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

```
typedef struct
{
      u16      UdpPort;
      u16      UdpTxDelay;
      ascii    UdpServ[UDPSOCK_STRMAXSIZE+1];
}ed_UdpSocketSetupParams_t;
```

- **UdpPort:**
  This parameter holds:
  - the remote peer UDP port used for a "client" UDP session (i.e. the IP Connectivity library user may be the first to transmit to the UdpServ / UdpPort address)

    OR
  - the local listening UDP port for a "server" UDP session (i.e. the IP Connectivity library opens a local port in listen mode and waits for a remote peer, filtered by the UdpServ parameter, to send a first datagram).

  By default, this parameter is set to 0.

  SYNTAX:

  1 to 5 numeric digits (between 0 and 9). Numbers above 65,535 are illegal as the port identification fields are 16 bits long in the TCP header. Port number 0 is also illegal.

- **UdpTxDelay:**
  This parameter holds the delay (expressed in ms) before the socket flushes the characters to transmit to the remote entity.

  The IP Connectivity library user provides data to transmit to the remote by using the ed_SendData() function (see the pfDataRequest field of the ed_SocketUDPStart() function for information on the full mechanism). Suppose the IP Connectivity library user calls ed_SendData() and provides it with data Byte by byte. Internally, the IP Connectivity library will not immediately send the byte since this would congest the internet network. It will "wait" until it has a "reasonable" amount of Bytes to transmit. However, if the IP Connectivity library user does not give more data for **UdpTxDelay** ms, the IP Connectivity library will transmit data already provided.

  By default, this parameter is set to 100 ms.

  SYNTAX:

  [0, 32760] in 20 ms steps.

- **UdpServ:**
  For a "client" UDP session (the IP Connectivity library user may be the first to transmit to the UdpServ/UdpPort address), this parameter contains the

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

address of the remote UDP peer that should be used. The “255.255.255.255” special IP address is forbidden.
For a “server” UDP session (the IP Connectivity library opens a local port in listen mode and waits for a remote peer to send a first datagram), this parameter is used to apply filtering of incoming UDP datagrams from a remote destination IP address. Only datagrams from the configured IP address will be allowed to connect to the Wavecom module. In case of no filtering the UdpServ parameter must be set to the "255.255.255.255" value. “0.0.0.0” is an invalid IP address for both “client” and “server” configuration.

SYNTAX:

32-bit number in dotted-decimal notation (i.e. xxx.xxx.xxx.xxx)

or alphanumeric ASCII text string up to UDPSOCK_STRMAXSIZE characters if DNS is integrated.

By default, this parameter is set to empty string.

2.7.3 The ed_UdpSocketSetConfig function

This function sets up some specific parameters about the UDP Socket management.

2.7.3.1 Prototype

```
s8 ed_UdpSocketSetConfig(ed_UdpSocketSetupParams_t *Params);
```

2.7.3.2 Parameters

Params:
Structure containing the parameter to set up. This structure is defined above.

2.7.3.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NOT_INIT
- ED_ERR_PARAM_BAD_VALUE
- ED_ERR_PARAM_TOO_LONG
- ED_ERR_PARAM_WRITE_E2P
- ED_ERR_INVALID_ARG
- ED_ERR_SERVICE_ALREADY_RUNNING

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.7.4 The ed_UdpSocketGetConfig function

This function gets up some specific parameters about the UDP Socket management.

2.7.4.1 Prototype

```
s8 ed_UdpSocketGetConfig(ed_UdpSocketSetupParams_t *Params);
```

2.7.4.2 Parameters

Params:
Structure containing the parameter to get. This structure is defined above.

2.7.4.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG

See §3 - List of error codes and reply codes for more information on error codes.

2.7.5 The ed_SocketUDPStart function

This function is used to create a UDP session in active mode or in listen mode, if the device is connected.

Once an IP link is established, the attached application can initiate a UDP session at any time (except if one of the IP Connectivity library service is already in use).

If this function is used to start a listening UDP socket (**`Listen`** parameter equal to 1), the UDP session will be active upon reception of a first datagram sent by a remote allowed UDP peer (**`UdpServ`** used as IP filter) on the appropriate local UDP port (**`UdpPort`**).

If this function is used to start an active UDP session(also called client session), the UDP session is effective as soon as the IP Connectivity library replies that the UDP session was successfully started.

Note that an effective UDP session (datagrams have already been exchanged) can only be stopped by the IP Connectivity library user (using the eD_SendData() function).

Note that a UDP session that is in listen mode can be stopped using the ed_SocketLUDPStop() function.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.7.5.1 Prototype

```
s8 ed_SocketUDPStart ( u8 Listen,
                       ed_ResponseCbk_f pfResponseCbk,
                       ed_DataHandler_f pfDataHnd,
                       ed_DataRequest_f pfDataRequest);
```

2.7.5.2 Parameters

Listen:

- Set to 1 if a listening UDP socket is to be started. UdpPort is the local port number and it must not be 0(zero). UdpServ is used as an IP address filter. The UDP session will be effective upon reception of the first datagram.
- Else if an active UDP session is to be started. UdpPort is the remote port number and it must not be 0(zero). Udpserv is the remote peer IP address.

pfResponseCbk:

pfResponseCbk is called by the IP Connectivity library when the Socket UDP session is started or when an error occurred.

This callback function is defined as below:

```
typedef void (* ed_ ResponseCbk_f) (s32 ResponseCode,
                                    TeDHandle id);
```

TeDHandle id: always set to ED_ID_UDPSOCKET.

ResponseCode values:

- ED_INFO_LISTEN_SET
- ED_OK_UDP_CLOSED
- ED_INFO_WAITING_FOR_DATA
- ED_ERR_DISTANT_DNS
- ED_ERR_STACK_INTERNAL
- ED_ERR_LISTEN_STOP
- ED_ERR_NETWORK_KO
- ED_ERR_GPRS_SESSION_LOST

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

pfDataHnd:

pfDataHnd is called by the IP Connectivity library to forward data received from the remote, to the application.

When the socket is closed, IP Connectivity library calls pfDataHnd with Data pointer set to NULL and DataLen set to 0.

This callback function is defined as below:

```
typedef u16 (* ed_DataHandler_f ) (u16 DataLen, u8 * Data,
                                   TeDHandle id);
```

where:

- **u16 DataLen**: number of available Bytes in u8 *Data.
- **u8 * Data**: Data pointer; freed by eDlib.
- **TeDHandle id**: always set to ED_ID_UDPSOCKET.
- **::**RETURN: Number of bytes "consumed" by the callback. Data left will be handed over at the next call, possibly with additional data.

pfDataRequest:

pfDataRequest is called by the IP Connectivity library to inform the application that it can send new data. It is called as soon as the UDP session is effective, to initiate the data transfer.

This callback function is defined as below:

```
typedef void (* ed_DataRequest_f) (u16 MaxLen, TeDHandle id);
```

where

TeDHandle id: always set to ED_ID_UDPSOCKET.

MaxLen is the maximum number of bytes that can be sent now (using the ed_SendData() function (c.f. section 2.9.1 below)).

2.7.5.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG
- ED_ERR_STACK_BAD_CONFIG
- ED_ERR_PHY_NOT_ACTIVATED
- ED_ERR_STACK_BUSY

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.7.6 The ed_SocketLUDPStop function

This function allows to close a UDP socket in listening mode (previously launched by the ed_SocketUDPStart() command).

2.7.6.1 Prototype

```
s8 ed_SocketLUDPStop (ed_ResponseCbk_f pfResponseCbk);
```

2.7.6.2 Parameters

pfResponseCbk:

pfResponseCbk is called by the IP Connectivity library when the Socket UDP is closed or when an error occurred.

This callback function is defined as below:

```
typedef void (* ed_ ResponseCbk_f) (s32 ResponseCode,
                                    TeDHandle id);
```

TeDHandle id: always set to ED_ID_UDPSOCKET.

ResponseCode value:

- ED_OK_UDP_CLOSED

2.7.6.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG
- ED_ERR_NO_LISTEN

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.8 PING Service

The IP Connectivity library provides this service to issue PING echo requests, manage the corresponding echo replies, and setup PING parameters.

2.8.1 Required Header Files.

The interface header file for the PING service is:

ed_ping.h

2.8.2 The ed_PINGSetupParams_t type

Structure containing the parameters about the PING management.

```
#define PING_STRMAXSIZE     120


typedef struct
{
      u8        PingDelay;
      ascii     PingRemote[PING_STRMAXSIZE+1];
}ed_PINGSetupParams_t;
```

- PingDelay:
  This parameter holds the maximum delay, in seconds, for which the ping task will wait for an echo reply. If this delay expires, the echo is considered as not answered and the ping task exits.

  SYNTAX:

  Integer in the range [0, 255]. Default value is 1.

- PingRemote:
  This parameter holds the address of the host to ping.

  SYNTAX:

  32-bit number in dotted-decimal notation (i.e. xxx.xxx.xxx.xxx)

  or alphanumeric ASCII text string up to PING_STRMAXSIZE characters if DNS is integrated.

  By default, this parameter is set to an empty string. Note that “0.0.0.0” is an invalid IP address.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.8.3 The ed_PINGSetConfig function

This function sets up some specific parameters about the PING management.

2.8.3.1 Prototype

```
s8 ed_PINGSetConfig(ed_PINGSetupParams_t *Params);
```

2.8.3.2 Parameters

Params:
Structure containing the parameter to set up. This structure is defined above.

2.8.3.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NOT_INIT
- ED_ERR_PARAM_BAD_VALUE
- ED_ERR_PARAM_TOO_LONG
- ED_ERR_PARAM_WRITE_E2P
- ED_ERR_INVALID_ARG
- ED_ERR_SERVICE_ALREADY_RUNNING

See §3 - List of error codes and reply codes for more information on error codes.

2.8.4 The ed_PINGGetConfig function

This function gets the parameters of the PING management.

2.8.4.1 Prototype

```
s8 ed_PINGGetConfig(ed_PINGSetupParams_t *Params);
```

2.8.4.2 Parameters

Params:
Structure containing the parameter to get. This structure is defined above.

2.8.4.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.8.5 The ed_PingEchoRequest function

If the device is connected, this function issues one echo request to the PingRemote host and start a task waiting for the echo reply. The echo request is made of 40 ICMP bytes (32 data bytes + 8 ICMP header bytes).

The ÏNG service returns ED_OK_ECHO_REPLY and exits if the echo reply is received within PingDelay seconds. The service returns ED_ERR_NO_REPLY and exits if PingDelay expired or if the ed_PingReset() function was called.

Note that the PING task can be stopped using the ed_PingReset() function.

2.8.5.1 Prototype

```
s8 ed_PingEchoRequest (ed_ResponseCbk_f pfResponseCbk);
```

2.8.5.2 Parameters

pfResponseCbk:

pfResponseCbk is called by the IP Connectivity library when a reply is received, or if the delay expired, or when an error occurred.

This callback function is defined as below:

```
typedef void (* ed_ ResponseCbk_f) (s32 ResponseCode,
                                    TeDHandle id);
```

TeDHandle id: always set to ED_ID_PING.

ResponseCode values:

- ED_INFO_ECHO_REQUEST
- ED_OK_ECHO_REPLY
- ED_ERR_DISTANT_DNS
- ED_ERR_STACK_INTERNAL
- ED_ERR_NO_REPLY
- ED_ERR_NETWORK_KO
- ED_ERR_GPRS_SESSION_LOST

2.8.5.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG
- ED_ERR_STACK_BAD_CONFIG
- ED_ERR_PHY_NOT_ACTIVATED
- ED_ERR_STACK_BUSY

See §3 - List of error codes and reply codes for more information on error codes.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

2.8.6 The ed_PingReset function

This is a two purposes function:

1. It resets the ICMP seqnum header field to 0. In a context of a PING application, it is relevant to call this function before issuing the first call to ed_PingEchoRequest(). Everytime a Ping echo request is sent, the ICMP seqnum (sequence number) is incremented.
2. It stops the PING task if it is active.

2.8.6.1 Prototype

```
s8 ed_PingReset(void);
```

2.8.6.2 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NOT_INIT

See §3 - List of error codes and reply codes for more information on error codes.

2.9 Data Flows Management

IP connectivity library provides this service to handle the transmission data flows.

2.9.1 Function ed_SendData

This function is used to send data with the TCP/IP stack.

2.9.1.1 Prototype

```
s8 ed_SendData (u8 *Data, u16 DataLen, bool bEnd);
```

2.9.1.2 Parameters

u8 *Data

Pointer on the data to send.

Note: Data pointer must be freed by the application

u16 DataLen

Size of the data to send. It must be less or equal to **MaxLen** parameter given with the previous ed_DataRequest_f callback call.

bool bEnd

When set to TRUE, this flag is used to indicate to the IP connectivity library that it receives the last block of data to send for the current

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

session. Once this boolean indicator is set, the IP connectivity library will not request data anymore.

2.9.1.3 Returned values

This function returns 0 on success, or a negative error value.
Possible error values are:

- ED_ERR_NOT_INIT
- ED_ERR_INVALID_ARG
- ED_ERR_INVALID_CALL
- ED_ERR_INVALID_SIZE

See §3 - List of error codes and reply codes for more information on error codes.

2.9.2 Function ed_SendDataExt

This function is used to send data with the TCP/IP stack for the TCP socket service only.

Notes:

- "Ext" stands for Extended.
- As long as the appropriate id is set, this function may be used with these other following services: SMTP(see 2.4.1.12), FTP(see 2.5.12), UDP socket (see 2.7.5).

- **Prototype**

```
s8 ed_SendDataExt(u8 *Data, u16 DataLen, bool bEnd, TeDHandle id);
```

- **Parameters**

u8 *Data: (IN)

Pointer on the data to send.

Data pointer must be freed by the application

u16 DataLen: (IN)

Size of the data to send. Must be ≤ MaxLen parameter given with the previous ed_DataRequest_f callback call (c.f. sections 2.4.1.12, 2.5.12 and 2.7.5).

bool bEnd: (IN)

When set to 1, flag to tell the eDlib that it is the last block of data to send for the current session. Once this bool is set, the eDlib won't request data anymore.

TeDHandle id: (IN)

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

Service id; one of: ED_ID_SMTP_PUTMAIL, ED_ID_FTP_PUT, ED_ID_TCPSOCKET_1, ED_ID_TCPSOCKET_2, ED_ID_TCPSOCKET_3, ED_ID_TCPSOCKET_4, ED_ID_TCPSOCKET_5, ED_ID_TCPSOCKET_6, ED_ID_TCPSOCKET_7, ED_ID_TCPSOCKET_8, ED_ID_TCPSOCKET_9, ED_ID_TCPSOCKET_10, ED_ID_UDPSOCKET.

- **Returned values**

  This function returns 0 on success, or a negative error value.
  Possible error values are:
  - *if the TCPIP service is not supported by the product.*
  - ED_ERR_NOT_INIT
    The eDLib cannot be used because its init function ed_Init() has not been called yet.
  - ED_ERR_INVALID_ARG
    in case Data is a NULL pointer.
  - ED_ERR_INVALID_CALL
    This function cannot be called now (because no matching ed_DataRequest_f callback call was issued or because no eDLib service is currently active).
  - ED_ERR_INVALID_SIZE
    The DataLen parameter is greater than expected.
  - ED_ERR_INVALID_ID
    The id parameter is invalid.
  - ED_ERR_ID_INACTIVE

  The service represented by the id is currently inactive.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

3 List of error codes and reply codes

Codes	Numeric Value	Description
ED_ERR_NO_HIGH_LEVEL_INTERNET	-1	High level Internet is not activated in the WISMO product.
ED_ERR_TCPIP_ALREADYUSED	-2	Another application is already using IP connectivity library
ED_ERR_PARAM_WRITE_E2P	-3	Parameter writing into E2P failed
ED_ERR_PARAM_READ_E2P	-4	Parameter reading from E2P failed
ED_ERR_PARAM_BAD_VALUE	-5	Parameter value is invalid
ED_ERR_PARAM_TOO_LONG	-6	Parameter string is too long
ED_ERR_INVALID_ARG	-7	Argument is invalid (or NULL pointer)
ED_ERR_MODEM_RUNNING	-8	WISMO has already a physical connection active. IP connectivity library can not be used
ED_ERR_INVALID_RING_TYPE	-9	RING is a Voice ring. Voice incoming calls are not accepted by IP connectivity library
ED_ERR_PIN_NOT_ENTERED	-10	PIN code has not been entered by user
ED_ERR_PHY_NOT_ACTIVATED	-11	The IP connectivity library network connection is not active
ED_ERR_STACK_BAD_CONFIG	-12	IP connectivity library application has an invalid configuration
ED_ERR_STACK_BUSY	-13	Another IP connectivity library application is already running
ED_ERR_NETWORK_KO	-14	WISMO module is not attached to the network. *Network was lost before the requested end of connection.*
ED_ERR_SIM_REMOVED	-15	SIM card is removed *before the requested end of connection.*
ED_ERR_ON_HOOK	-16	internal On-Hook command returns ERROR
ED_ERR_BUSY	-17	A busy signal is detected on the remote site
ED_ERR_NO_CARRIER	-18	The modem handshaking process with the remote host is interrupted or unsuccessful
ED_ERR_PPP_ERROR	-19	The PPP negotiation failed *(check ISPUN, ISPPW (see §2.3.2.2) and PPPMODE (see §2.3.3.2))*

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

Codes	Numeric Value	Description
ED_ERR_PPP_TIMEOUT	-20	PPP negotiation timeout expired
ED_ERR_INVALID_EVENT	-21	An invalid event occurred during physical activation. The activation is aborted
ED_ERR_PHY_TIMEOUT	-22	Inactivity timeout expired. Physical connection is closed
ED_ERR_GPRS_ABORTED	-23	An internal GPRS command error occurred. The activation is aborted
ED_ERR_WM_UNABLE_SUBS_EVENT	-24	Subscription of command, timer or unsolicited message failed
ED_ERR_WM_FLOW_OPEN	-25	Unable to open internal data flows
ED_ERR_WM_FLOW_CLOSE	-26	Unable to close internal data flows
ED_ERR_WM_PARSER_FLOW	-27	Timeout on reception of flows event
ED_ERR_WM_GPRS_OPEN	-28	Open GPRS internal session failed
ED_ERR_WM_GPRS_AUTH	-29	GPRS authentication failed
ED_ERR_WM_GPRS_IPCP	-30	Error when IP connectivity library get GPRS IPCP data
ED_ERR_STACK_INTERNAL	-31	IP connectivity library internal error
ED_ERR_DISTANT_DNS	-32	Remote server name has not been mapped by DNS. TCP/IP stack is not able to reach the primary and secondary DNS servers or a wrong server address has been filled in.
ED_ERR_DISTANT_NO_RESP	-33	No response from remote server.
ED_ERR_DISTANT_OPEN	-34	Error during IP connectivity library application session opening
ED_ERR_DISTANT_CLOSE	-35	Error during IP connectivity library application session closing
ED_ERR_DISTANT_CCREC_ADDR	-36	One of the CCREC e-mail addresses is rejected by remote SMTP server
ED_ERR_DISTANT_REC_ADDR	-37	One of the REC e-mail addresses is rejected by remote SMTP server
ED_ERR_DISTANT_SENDER_ADDR	-38	Sender e-mail address(es) is rejected by remote SMTP server
ED_ERR_DISTANT_BODY	-39	e-mail body to send is rejected by remote SMTP server
ED_ERR_DISTANT_SEND	-40	Data sending attempt failed
ED_ERR_DISTANT_TCP_CLOSED	-41	TCP session is closed by IP connectivity library

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

Codes	Numeric Value	Description
ED_ERR_DISTANT_TCP_CLOSED_BY_PEER	-42	TCP session is closed by remote server
ED_ERR_DISTANT_USERNAME	-43	Username is rejected by remote SMTP server
ED_ERR_DISTANT_PASSWORD	-44	Password is rejected by remote SMTP server
ED_ERR_DISTANT_DATA_RETR	-45	Remote server rejected data retrieval request
ED_ERR_DISTANT_EMAIL_RETR	-46	e-mail retrieval failed
ED_ERR_DISTANT_EMAIL_HDR_RETR	-47	e-mail header retrieval failed
ED_ERR_DISTANT_DEL	-48	Deletion of e-mail rejected by remote server
ED_ERR_DISTANT_CD	-49	Change directory on remote server failed
ED_ERR_INVALID_SIZE	-50	The DataLen parameter of the ed_SendData() or ed_SendDataExt()function is greater than expected.
ED_ERR_INVALID_CALL	-51	The ed_SendData() or ed_SendDataExt() function can not be called now (no matching ed_DataRequest_f callback call was issued or because no IP connectivity library service is currently active).
ED_ERR_NOT_INIT	-52	The IP connectivity library can not be used because its init function ed_Init() has not been called yet.
ED_ERR_LISTEN_STOP	-53	The listening TCP/UDP socket has been canceled.
ED_ERR_NO_LISTEN	-54	ed_SocketLTCPStop() /ed_SocketLUDPStop()has been called while no TCP/UDP socket was opened in listen mode.
ED_ERR_NO_GPRS	-55	GPRS feature unavailable.
ED_ERR_NO_ANSWER	-56	There is no response from the remote site
ED_ERR_CLI_INVALID	-57	CallerID does not provide caller phone number or it is invalid
ED_ERR_NO_REPLY	-58	No echo reply received (PING service).
ED_ERR_PARAM_BAD_MAILIDX	-59	E-mail address index is invalid.
ED_ERR_GPRS_SESSION_LOST	-60	The GPRS session is lost(context deactivated).

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

Codes	Numeric Value	Description
ED_ERR_SERVICE_ALREADY_RUNNING	-61	It is not possible to set new parameter value because the associated service is already in use.
ED_ERR_INVALID_ID	-62	The id parameter is invalid.
ED_ERR_ID_INACTIVE	-63	The service represented by the id is currently inactive.
ED_ERR_MEMALLOC_FAILED	-64	Not enough memory left to run a new socket.
ED_ERR_SOCKET_CREATION_BUSY	-65	There can only be one socket creation at a time.

Codes	Numeric Value	Description
ED_INFO_CONNECT	100	Modem is connected
ED_INFO_WAITING_FOR_DATA	107	IP connectivity library is waiting for data to send
ED_INFO_MAIL	108	Start of the e-mail data retrieval from the mailbox
ED_INFO_NOMAIL	109	No new e-mail in the mailbox
ED_INFO_DATA_BEGIN	110	Start of the data retrieval
ED_INFO_LISTEN_SET	111	The listening TCP socket has been successfully created. It is now waiting for a remote entity to connect.
ED_INFO_ECHO_REQUEST	112	The PING echo request is being sent.

Codes	Numeric Value	Description
ED_OK_PPP	1	PPP negotiation is OK. As soon as the TCP/IP stack software displays this message, it is ready to receive commands.
ED_OK_GPRS_SESSION_SET	2	GPRS session is activated
ED_OK_ON_HOOK	3	Connection is correctly released
ED_OK_MAIL_SENT	4	ed_SendMail or ed_PutMail has successfully completed
ED_OK_TCP_CLOSED	5	ed_SocketTCPStart has successfully completed. The TCP Socket is closed correctly.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

Codes	Numeric Value	Description
ED_OK_POP3	6	ed_GetMail has successfully completed. This response is issued when the POP3 session is gracefully closed.
ED_OK_FILE_TRANSFERED	7	ed_FTPGet or ed_FTPPut have successfully completed. The FTP transfer file has finished successfully.
ED_OK_UDP_CLOSED	8	The UDP socket was correctly closed.
ED_OK_ECHO_REPLY	9	Echo reply received (PING service).

Codes	Numeric Value	Description
ED_ENTRY_LEVEL_TCP	0	Entry level TCP/IP configuration detected (Only socketTCP, socketUDP and ping applications are available).
ED_HIGH_LEVEL_INTERNET	1	High level TCP/IP configuration detected (socketTCP, socketUP, ping, FTP and email applications are available).

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

4 Appendix: Open AT® Resources

Constraints about the use of common resources for the whole Open AT® software are expressed hereafter.

4.1 Flash objects

Open AT® IP Connectivity library may uses up to 170 Flash objects

4.2 Timers

5 timers are used by the IP Connectivity library.

4.3 RAM and ROM occupation

RAM :

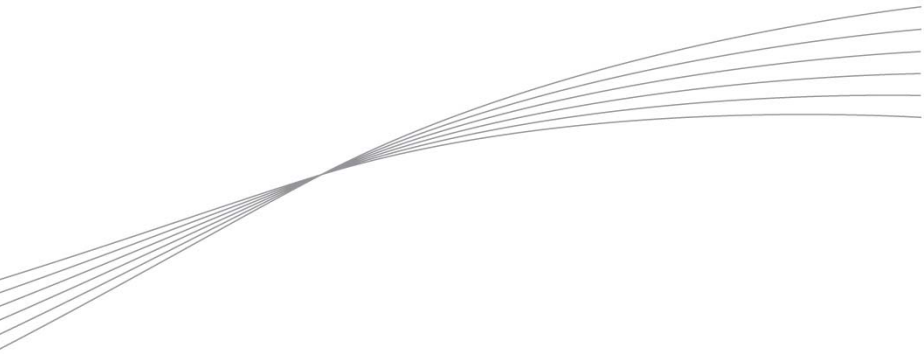
- 7551 bytes statically allocated.
- 30720 bytes (B memory) or 7344 bytes (A memory) dynamically at boot time.
- Memory dynamically allocated by each application

FTP Put	53 bytes
FTP Get	1847 bytes
SMTP	53 bytes
POP3	1847 bytes
each Socket TCP	2029 bytes
socket UDP	1896 bytes

ROM : 57252 bytes

Note: these values do not include Open AT® or ADL libraries resource allocation.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.



wavecom

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