

LHA

Telex Tester

SPECIFICATION FOR
API TYPE 60

LHA TelexTester

**LHA TelexTester
Lucknow House
1A St Leonard's Road
Surbiton
Surrey, KT6 4DE
England**

Telephone (44) 181 390 4200
Fax (44) 181 399 6227

The information contained in this manual is proprietary and shall be utilized only to produce software that uses TelexTester and shall not be used for any other purposes. LHA Limited owns the copyright of this document (and the information contained herein) which is supplied in confidence and must not be used for any purpose other than that for which it is supplied and must not be reproduced without permission in writing from the owners.

LHA MAKES NO WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THIS PROPRIETARY INFORMATION.

Version 6
SPECIFICATION FOR API TYPE 60

Documentation date 1st January 1999

All rights reserved. No part of the contents of this document may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

CONTENTS

INTRODUCTION	7
COMPONENTS	8
CONFIGURATION	10
CALL PARAMETERS	11
CALL CONVENTION	12
FUNCTIONS	12
GENERAL.....	12
STRUCTURE OF LPERROR	13
General	13
Format 1 of lpError	13
Format 2 of lpError	14
STRUCTURE OF LPAPI.....	14
FUNCTION 1, BUILD DATA FILES	14
FUNCTION 2, GET DATA FILE VERSION.....	15
FUNCTION 3, GET LICENCEE NAME	16
FUNCTION 4, ENCODE TESTKEY	16
FUNCTION 5, DECODE TESTKEY	16
FUNCTION 6, USER VALIDATION.....	17
FUNCTION 7, GET OUTGOING MESSAGE	17
FUNCTION 8, GET INCOMING MESSAGE.....	18
FUNCTION 9, GET API60.DLL VERSION	19
USER PROFILE	20
GENERAL.....	20
RA7, USER NAME	20
RA8 (USER PROFILE).....	20
RA9 (MAXIMUM TEST VALUE)	21
RAA (VERSION DETAILS)	21
RAB (PARAMETERS)	22
MESSAGE DETAILS	23
GENERAL.....	23
RB1 MESSAGE DETAILS.....	23
RB2 MESSAGE DETAILS.....	24
32 CURRENCY & AMOUNTS	25
CHR CHARACTER TABLE.....	25

RAI ANALYSIS.....	25
TESTKEY PARAMETER BLOCK (TPB).....	27
GENERAL.....	27
TPB TAGS.....	27
General.....	27
USR (User tag).....	27
PW (Password tag).....	27
COR (Correspondent tag).....	28
30 (Date tag).....	28
15 (Testkey tag).....	28
RES (Test result tag).....	28
32 (Amount & Currency tag).....	28
MSG (Message number).....	29
SC (Manual value).....	29
CHR (Character table).....	29
TESTKEY RESULTS.....	31
ERROR CODES.....	33
FILE HANDLING ERRORS.....	33
ERROR MESSAGES.....	34

Introduction

TelexTester has become a worldwide standard for Testkey calculations, therefore application developers require to access the TelexTester calculation system. To this end LHA has provided various Application Program Interfaces (APIs).

Documentation for specific APIs is available from LHA, this document details the operation of API Type 60. This API was developed in order that vendors could produce a graphical front end to the TelexTester system. The API is supplied in the form of a dynamic link library (DLL file). A single entry point in this library provides various functions.

Components

The following components are supplied with the standard TelexTester product:

File name	Purpose	Location
API60.dll	Dynamic link library with entry point 'LHAAPI'. This entry point is used for all function calls.	On Path
TT32.exe	This executable program launches the character mode TelexTester system.	On Path
TK32.lbr	Library of TelexTester support modules	With data files (or on DPATH)
MESSAGE2.DAT	Text of screen displays and other messages	With data files
CBLRTSS.dll CBLDWINS.dll CBLFHS.dll CBLINTS.dll MFFH.dll	COBOL run time support module	On Path
ADISCTRL	Keyboard configuration file	With data files
MENU2.MNU	Menu configuration	With data files
CURRENCY.ISO	Currency codes	With data files
TKUPD232.LBR	Data file conversion library, used to update data files between versions.	With data files (or on DPATH)
API60.EXE	Program to test the API 60.	On Path

Configuration

Environment variable 'TELEXTESER' is used to locate the data files, if not set this will default to the current directory.

Call parameters

The LHA API function in API60.dll is used for all functions.

LHA API (lpOption, lpAPI, lpError)

lpOption	Pointer to 32 bit integer containing function required
lpAPI	Pointer to 1920 byte buffer for other parameters and function return values.
lpError	Pointer to 80 byte buffer for error messages

The buffers at pAPI and lpError must be of the size specified for all function calls.

Results will be returned in the lpAPI parameter.

lpError will only be used for abnormal error results or when the :RES= tag is not present in the lpAPI parameter.

For example, if an invalid password is supplied to the Testkey calculation function, this will be reported in the :RES= tag of the lpAPI parameter. However, if this tag is not available the error will be reported in the lpError parameter.

In addition, errors are added by the TelexTester system to the text file SYSLOG.TXT in the Data files directory.

Call convention

Call convention is cdecl.

Functions

General

All functions require the three parameters to be passed.

All functions require the file 'MESSAGE2.DAT' to be present in the data directory.

All functions will return lpError as space if successful, otherwise, lpError will give an indication of the problem.

TelexTester writes error messages to the file SYSLOG.TXT in the data directory.

The LHA API works with fixed length arrays of bytes, both lpError and lpAPI must be terminated by the calling program with a null if null strings are required.

The lpAPI buffer lines should be padded with spaces, not null terminated.

Structure of lpError

General

On output, if lpError is not set to spaces then an error has occurred.

lpError can have two formats:

Format 1 of lpError

nnn*xxxxxxx

Where:

nnn is a three digit number

xxxxx is the description of the result

See the Error Codes section for possible errors.

Format 2 of lpError

Structure of lpAPI

This parameter points to a 1920 byte area.

This area is divided into 24 lines of 80 bytes.

Most functions required multiple parameters to be passed to the TelexTester system, a single parameter is placed at the start of each line with a tag followed by the parameter value, eg:

:USR=STEPHEN

Some functions, eg Function 1, require no parameters, however, the lpAPI parameter must still be included.

Some functions, eg Bank name and Version will return their single value at the start of the lpAPI buffer with no tag preceding.

Function 1, Build data files

Used to build the Telextester data files. This will only work if the file GLC.DAT is not found.

All data files are built by this function.

Function 2, Get Data file Version

Get version information, returns the version information from the Telextester data files, eg:

V6.3.5 API61 V001

In the form:

Vx.y.z API61 Vnnn

Where:

x is the major version, currently 6

y is the minor version currently 3

z is the sub version currently 5

nnn is the API version currently 001

z changes are program modifications

x or y change to indicate structure changes in the data files.

Function 3, Get Licencee name

Returns the licenced bank name.

Function 4, Encode Testkey

This function uses the pAPI parameter as the Testkey Parameter Block (TPB). See TPB section for details.

Following successful calculation of the testkey, this function automatically calls function 7, Get Outgoing Message and returns the details as specified in 'Message Details'.

Function 5, Decode Testkey

This function uses the pAPI parameter as the Testkey Parameter Block (TPB). See TPB section for details.

Following verification of the testkey, whether successful or not, this function automatically calls function 8, Get Incoming Message and returns the details as specified in 'Message Details'.

Function 6, User validation

This function checks that the user id and password passed in the lpAPI parameter are valid, it also returns information about the user, see User Profile.

The function requires three components in the lpAPI parameter:

:USR=

:PW =

:RES=

If the function is successful, the :RES= parameter will be set to space and the User Profile fields will be completed.

Function 7, Get Outgoing Message

This function is used to retrieve the details for a specific outgoing message, which has been previously tested by the TelexTester system.

The function requires two components in the lpAPI parameter:

:MSG=

:RES=

If the function is successful, the :RES= parameter will be set to one of the values specified in 'Testkey results' otherwise this tag will hold the error details.

If successful, the Message Detail fields will be completed. See 'Message Details' later for the returned information.

The :MSG= tag must be in the form:

:MSG=nnnnn

Where nnnnn are all numeric digits.

Function 8, Get Incoming Message

This function is used to retrieve the details for a specific incoming message, which has been previously validated by the TelexTester system.

The function requires two components in the lpAPI parameter:

:MSG=

:RES=

If the function is successful, the :RES= parameter will be set to one of the values specified in ‘**Testkey results**’ otherwise this tag will hold the error details.

If successful, the Message Detail fields will be completed. See ‘Message Details’ later for the returned information.

Function 9, Get API60.DLL Version

Get version information from the dll file API60.DLL. This function can be called to verify that the correct dll is being loaded eg:

V6.3.5

In the form:

Vx.y.z

Where:

x is the major version, currently 6

y is the minor version currently 3

z is the sub version currently 5

z changes are program modifications

x or y change to indicate structure changes in the data files.

User Profile

General

The User Profile is returned in the lpAPI parameter following successful User Validation, function 6.

The following tags and values are returned:

RA7, User Name

The user name is returned.

RA8 (User profile)

The following string is returned:

```
RA8=abcdefg
```

Where:

a = <Y> if allowed to run Variables report

b = <Y> if allowed to analyse testkey results

c = The operator status, 'S', 'O' or 'V'

d = <Y> if allowed to do outgoing tests

e = <Y> if allowed to do incoming tests

f = <Y> if allowed to run the Start of day reports

RA9 (Maximum test value)

The maximum test value the user is allowed, or zero.

RAA (Version details)

The following TelexTester parameters are returned:

```
RAA=aaaaaabcd
```

Where:

aaaaaa = The version of TelexTester.

b = <Y> if the system parameters are set for dual control, requiring verification before a testkey is produced.

c = <Y> if the system parameters are set to make API tests verified automatically.

d = The next 40 characters are the bank name.

RAB (Parameters)

The following TelexTester parameters are returned:

RAB=abcde

Where:

a = <Y> if the system parameters are set for dual control, requiring verification before a testkey is produced.

b = <Y> if the system parameters are set to make API tests verified automatically.

c = <Y>, <1> or <2> from system parameters giving the method of verify input.

d = <Y> if same user is allowed to verify his tests.

e = <M> or <S> for a multi or single user system.

Message Details

General

The Message Details are returned in the lpAPI parameter following successful Incoming or Outgoing test retrieval, functions 7 and 8.

The message details are also returned following successful Incoming and Outgoing testkey calculations, functions 4 and 5, but only if the :MSG= tag is included.

The following tags are returned:

RB1 Message details

The following information is in the data:

:RB1=aaaaabbbbbbcdeffffggiiiiiiiijjjjmm

Where:

aaaaa is the message number

bbbbbb is the log date of the message, format 'YYMMDD'

Fixed number total

Date table total

Currency total

Amount total

Character table total

Manual value

Sequence table total

This information will not be included if the user is not allowed the analysis option through their profiles or if the analysis is not available, for example in the case of a zero testkey.

Testkey Parameter Block (TPB)

General

Any control characters (less than <SPACE>) present in the TPB are replaced with the space character.

TPB Tags

General

All tags are in the form:

:xxx=yyyy

Where xxx is a three character tag id and yyy is the value associated.

USR (User tag)

Indicates the user id.

PW (Password tag)

Indicates the user password.

COR (Correspondent tag)

Correspondent short code, up to 12 characters.

30 (Date tag)

Date of test, format always 'YYYYMMDD'.

15 (Testkey tag)

Testkey field, 22 characters long. Normally left justified but the Testkey system parameters allow lead zeroes to be applied on encoded testkeys.

RES (Test result tag)

This is the result of the testkey calculation in number and text form. First three characters are the message number followed by a '*' then the message text.

See the section **Testkey Results** for details.

32 (Amount & Currency tag)

Used to pass amount and currency information to and from the TelexTester system for Testkey calculation and Message details requests.

Example. :32 =USD 123456

The currency must be four characters followed by the amount.

Up to 14 of these tags can be present.

Comma separation is accepted on input to the API, however, the API will remove comma separation on output.

On output, the amount field will be padded to its maximum length with zeroes and given a decimal point and two decimal places. Therefore, the output length of this field is $4 + 16 + 1 + 2 = 23$ characters.

MSG (Message number)

Used to pass the message number from the TelexTester system back. This number is incremented for every test carried out. Separate number sequences are used for incoming and outgoing.

The message number must be five digits in length, eg:

:MSG=00012

SC (Manual value)

Used to pass a manual testkey component, up to seven digits.

Most correspondents do not use this component so it can be omitted. However, if it is required and not included an error will be returned in the 'RES' field.

CHR (Character table)

Used to pass character table entries from and to the Testkey system.

Most correspondents do not use this component, however, it can be included for every correspondent together with the data '\'. This '\' will be ignored unless the correspondent requires the CHR entry when an error will be returned.

Up to ten of these tags can be present with a total of 500 characters maximum.

Testkey Results

The result of testkey calculations, the Testkey value is placed in the :15= Tag.

The type of result is placed in the :RES= tag. This is in the format:

nnn*xxxxxxxx

Where:

nnn is a three digit number

xxxxx is the description of the result

The following values can appear in the RES tag:

249	Outgoing testkey successfully calculated
250	Incoming testkey successfully decoded
252	Incoming testkey successfully decoded Testkey had been previously missing
254	Incoming testkey successfully decoded Testkey was out of sequence causing missing testkeys
256	Incoming testkey, failed Testkey was incorrect
258	Incoming testkey, failed Testkey was a duplicate
Other	All other values are failures Reason for failure is included in message text.

Error codes

File handling errors

2-002	Duplicate record written. Attempt to write a record to a file which already exists	
9-014	Too many files open	Operating system file open limit exceeded. In DOS mode this is fixed by editing CONFIG.SYS/C ONFIG.NT and increasing files command.
2-003	Record not found	

Error messages

262	Error reading file or correspondent not found	The correspondent code is incorrect
264	API:Tags for Testkey & Results not found	Either or both of the tags: :15 = :RES= Are missing
42	Invalid amount format	
17	Invalid date	