CTIA Certification

A Division of CTIA–The Wireless Association™

Wireless Device High-Level API Specification

Revision 1.0
April 2005

CTIA Certification Program
1400 16th Street, NW, Suite 600
Washington, DC 20036

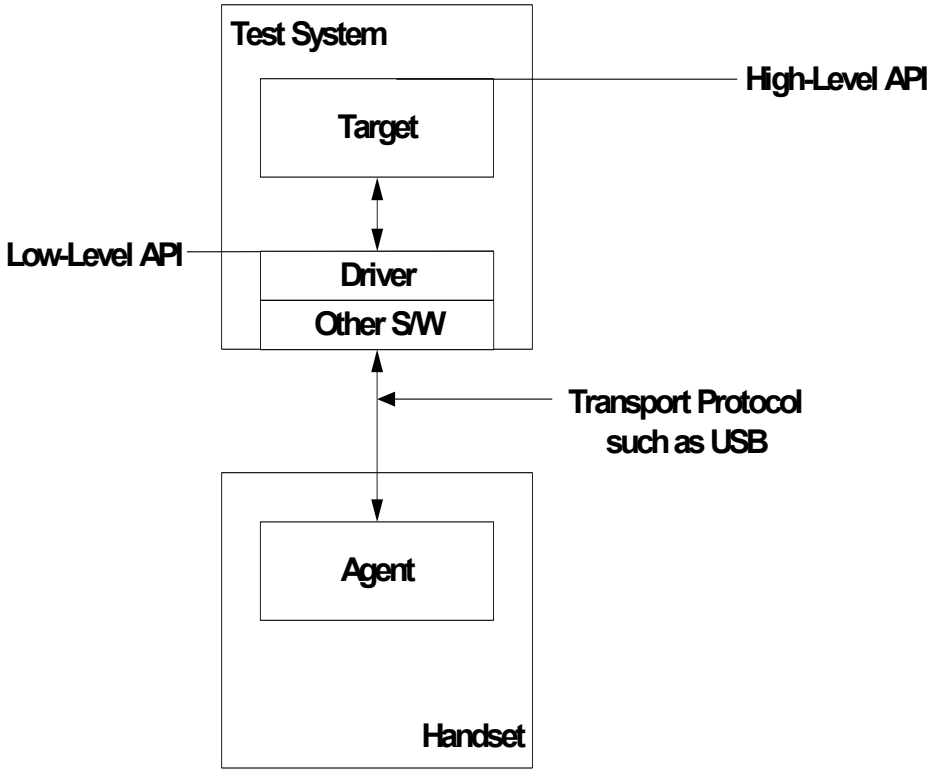
e-mail: certification@ctia.org
Telephone: 1.202.785.0081
www.ctia.org/certification

Table of Contents

1. Introduction

This document provides an interface definition for a high-level application-programming interface (API) between a mobile cellular phone (handset) and a test system or software program. This interface definition does not include transport protocol specification, leaving that particular implementation detail for designers. A separate document, Wireless Device Low-Level API Specification, defines the low-level API. [1]

The general block diagram of this system is shown below:



In the drawing depicted above, the “Driver” and the “Other S/W” are optional pieces of software that may or may not be present, depending on the implementation of the capability by the handset manufacturer. If they are present, they tend to make the LLAPI and the HLAPI appear to be convergent, however the primary distinction that needs to be drawn between the two is that the HLAPI is independent of handset / manufacturer, while the LLAPI is dependent.

[1] http://www.ctia.org/certification/best_practices/index.cfm

2. Abbreviations, Acronyms, & Definitions

Agent	The capability residing on the handset that implements the API. This capability may exist in either software or hardware
API	Application Programming Interface
DUT	Device Under Test
HLAPI	High Level Application Programming Interface
LED	Light Emitting Diode
LLAPI	Low Level Application Programming Interface
PC	Personal Computer
SMS	Short Message Service
Target	The software application residing outside the handset that communicates through the API with the Agent on the handset

3. API Definitions

3.1. ADD_PHONEBOOK_ENTRY

This API adds a phone book entry as specified by name and value parameters.

API: ADD_PHONEBOOK_ENTRY(type, name, value, kind, speed_no, ringer_id, icon)

Return Value: 0= success, 1 = failure, 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
type - null terminated character string Specifies the phone book entry type to be added	“Phone”	A phone number is to be added
	“Email”	An email entry is to be added
name - null terminated character string Specifies the name of the entry to be added	Varied	N/A
value - null terminated character string Specifies the value of the entry to be added	Varied	N/A
kind - null terminated character string Specifies in most cases the type of phone number entry such as "Home" or "Work" and etc, but can also be used to specify the type of email entry if apply. Assign null if not apply.	"WORK"	A work phone number or a work email is to be added.
	“HOME”	A home phone number or a home email is to be added.
	“MAIN”	A primary phone number or a primary email is to be added.
	“FAX”	A fax number is to be added. Only applies to type "Phone".
	“MOBILE”	A mobile phone number is to be added. Only applies to type "Phone".
	“PAGER”	A pager number is to be added. Only applies to type "Phone".

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
speed_no - null terminated character string Specifies the speed number associated with the entry.	Varied	The speedno string should contain only digits. It is not made an integer data type just to be consistent with EDIT_PHONEBOOK_ENTRY
ringer_id - null terminated character string Specifies the ID of the ringer.	Varied	N/A
icon - null terminated character string Specifies the file name or Id of the associated icon picture. Assign null if not apply.	Varied	N/A

3.2. COMPOSE_SMS_MESSAGE

Compose a SMS message on the mobile station as specified by the above parameters.

API: COMPOSE_SMS_MESSAGE(toaddr, mesg, callback, priority, ack-integer)
Return Value: 0= success, 1 = failure, 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
toaddr - null terminated character string Specifies the destination address. If multiple addresses are specified, they must be separated by a common.	Varied	N/A
mesg - null terminated character string Specifies the body of the message	Varied	N/A
callback - null terminated character string Specifies the callback number	Varied	N/A
priority - null terminated character string Specifies the message priority	"Normal"	The message should be marked normal
	"Urgent"	The message should be marked urgent
ack – integer Specifies the message delivery acknowledgement	1	The delivery acknowledgement needs to be sent to the message originator
	0	The delivery acknowledgement does not need to be sent to the message originator

3.3. DELETE_PHONEBOOK_ENTRY

Deletes a phone book entry as specified by type, name, kind and value parameters.

API: DELETE_PHONEBOOK_ENTRY(type, name, kind, value)
Return Value: 0= success, 1 = failure, 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
type – null terminated character string Specifies the phone book entry type to be searched	"Phone"	A phone number is to be searched
	"Email"	An email entry is to be searched
name - null terminated character string Specifies the name of the entry to be searched	Varied	N/A
kind – null terminated character string Specifies in most cases the type of phone number entry such as "Home" or "Work" and etc, but can also be used to specify the type of email entry if apply. Assign null if not apply.	"WORK"	Specifies the entry with an office phone number or a work email address if apply.
	"HOME"	Specifies a home phone number or a home email address if apply.
	"MAIN"	Specifies a primary phone number or a primary email.
	"FAX"	Specifies a fax number. Only applies to type "Phone".
	"MOBILE"	Specifies a mobile phone number. Only applies to type "Phone".
	"PAGER"	Specifies a pager number. Only applies to type "Phone".
value – null terminated character string Specifies the value of the entry to be searched	Varied	N/A

3.4. DIAL_NUMBER

This API depresses the keys necessary to input the specified number to the DUT, and then will perform the action necessary to initiate the call activity to the network (typically the "send" key on keypads, maybe similar area on systems with touchscreens). This API will only perform the stimulation necessary to perform these events, no verification activity is performed to ensure that the desired actions are actually initiated, and the API immediately returns control to the test case upon completing the stimulation. For instance, if it is necessary to verify that a call is initiated, the user would utilize another API to check for something like a "connecting" message / bitmap.

API: DIAL_NUMBER DIAL_NUMBER (sz number)
Return Value: 0= success, 1 = failure, 2-10 to be defined as needed.

API Variable / Variable Description	**API Allowable Parameters**	**API Parameter Description and Use**
sz number Any value. All values are interpreted to be numeric keys, and the system does not perform error checking on the input string.	Digits 0 to 9	N/A

3.5. EDIT_PHONEBOOK_ENTRY

API: EDIT_PHONEBOOK_ENTRY(type, name, kind, field, value)

Modify the specified field of a phone book entry as specified by type, name, kind and value parameters.

Return Value: 0= success, 1 = failure, 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
type - null terminated character string Specifies the phone book entry type to be searched	“Phone”	A phone number is to be searched
	“Email”	An email entry is to be searched
name - null terminated character string Specifies the name of the entry to be searched	Varied	N/A
kind - null terminated character string Specifies in most cases the type of phone number entry such as "Home" or "Work" and etc, but can also be used to specify the type of email entry if apply. Assign null if not apply.	"WORK"	Specifies the entry with an office phone number or a work email address if apply.
	“HOME”	Specifies a home phone number or a home email address if apply.
	“MAIN”	Specifies a primary phone number or a primary email.
	“FAX”	Specifies a fax number. Only applies to type "Phone".
	“MOBILE”	Specifies a mobile phone number. Only applies to type "Phone".
	“PAGER”	Specifies a pager number. Only applies to type "Phone".
field - null terminated character string Specifies the field of the entry to be modified.	"NAME"	Specifies that the name of the entry will be modified.
	“VALUE”	Specifies that the phone number or the email address of the entry will be modified.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
	"KIND"	Specifies that the phone type such as "WORK" of the entry will be modified. Can also be used to change the email type if apply.
	"SPEEDNO"	Specifies that the speed number of the entry will be modified.
	"RINGERID"	Specifies that the ringer of the entry will be modified.
	"ICON"	Specifies that the name or the ID of the icon picture associated with the entry will be modified
value - null terminated character string Specifies the new value of the specified field for the entry to be used.	Varied	

3.6. ENTER_TEXT

This API inputs the desired text string into the DUT. The text that will be entered is either input from a keyboard, virtual keyboard, or is run through a translation table (example, depressing the 2 key three times results in a 'c') to input the text. The DUT must be in a straight text entry mode when this API is used. If the DUT is in a predictive text entry mode, erroneous results will occur. The test entry mode is not detected as part of this API and the stimulation is simply performed prior to returning control to the test case.

API: ENTER_TEXT ENTER_TEXT(char * text)
Return Value: 0= success, 1 = failure, 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char * text The string representing the text that is to be entered.	Varied	Varied

3.7. ENTER_TEXT_IN_FIELD

This API will attempt to find the text label and a text field to the right of the text label, place the cursor / entry indicator in the text field and will then inputs the desired text string into the DUT. The text that will be entered is either input from a keyboard, virtual keyboard, or is run through a translation table (example, depressing the 2 key three times results in a 'c') to input the text. The DUT must be in a straight text entry mode when this API is used. If the DUT is in a predictive text entry mode, erroneous results will occur. The test entry mode is not detected as part of this API and the stimulation is simply performed prior to returning control to the test case.

API: ENTER_TEXT_IN_FIELD(char *szlabel, char *sztext)
Return Value: 0= success, 1 = failure (A value of 1 only if the text label cannot be found, or if a text field is not found to the right of the text label), 2-10 to be defined as needed.

API Variable / Variable Description	**API Allowable Parameters**	**API Parameter Description and Use**
char * szlabel The label associated with the text field in which the data is to be entered.	Varied	Varied
char * sztext The text that is to be entered into the text field	Varied	Varied

3.8. FIND_PHONEBOOK_ENTRY

This API finds a phone book entry as specified by type, name, kind and value parameters.

API: FIND_PHONEBOOK_ENTRY(type, name, kind, value)
Return Value: 0= success, 1 = failure, 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
type - null terminated character string Specifies the phone book entry type to be searched	“Phone”	A phone number is to be searched
	“Email”	An email entry is to be searched
name - null terminated character string Specifies the name of the entry to be searched	Varied	N/A
kind - null terminated character string Specifies in most cases the type of phone number entry such as "Home" or "Work" and etc, but can also be used to specify the type of email entry if apply. Assign null if not apply.	"WORK"	Specifies the entry with an office phone number or a work email address if apply.
	“HOME”	Specifies a home phone number or a home email address if apply.
	“MAIN”	Specifies a primary phone number or a primary email.
	“FAX”	Specifies a fax number. Only applies to type "Phone".
	“MOBILE”	Specifies a mobile phone number. Only applies to type "Phone".
	“PAGER”	Specifies a pager number. Only applies to type "Phone".

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
value - null terminated character string Specifies the value of the entry to be searched	Varied	N/A

3.9. HOLD_KEY

This API will press and hold the input key until a release is commanded. This API is typically paired in some fashion with RELEASE_KEY.
This API call is a non-blocking function, as opposed to PRESS_KEYS which is a blocking function. This API will perform the press of the desired key and return control back to the calling function in order to allow other actions to occur with the key held in the down state. The key will be held in the down state until a RELEASE_KEY call with the appropriate parameter is performed.

API: HOLD_KEY(*sz key)
Return Value: 0= success, 1 = failure, 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char * sz key The key that is to be held down until released.	Dependent on names of keys present on test target.	Dependent on names of keys present on test target.

3.10. PAUSE

This API waits the desired number of seconds before returning control to the test script level.

API: PAUSE(double intseconds)
Return Value: 0= success, 1 = failure, 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
double intseconds	Integer Values from 1 to 65535	This value is the number of seconds that will elapse until control is returned to the test script.

3.11. PRESS_BUTTON

This API will press a button on the handset. The button to be pressed can be specified as a .bmp to search for and locate, or a text string. This function is only used on touch screens, or visual displays where a cursor is used for control. For one handed devices this can be used for soft keys and for touch screen phones this can be used to tab a button.

API: PRESS_BUTTON(char *button, int ihold, int ibetween)
Return Value: 0= success, 1 = failure (A value of 1 is returned only if the button cannot be located), 2 10 to be defined as needed.

API Variable /Variable Description	API Allowable Parameters	API Parameter Description and Use
char * button Can be a .bmp to search for, or a text string to locate.	Varied	Specific parameters not currently defined.
int ihold This is the time in milliseconds to hold each key in the down state.	Integer Values from 1 to 65535	As specified
int ibetween This is the time in milliseconds to delay between the break of one key and the make of the subsequent key.	Integer Values from 1 to 65535	As specified

3.12. PRESS_KEYS

This API will press a key, a series of keys, or will perform a lookup on a named key and perform the desired keypress(es). This can be used to press a "softkey", where the name given to a softkey would be the same from handset to handset, but the location of the key may be different based on particular phone state. Example, the "menu" softkey may be left, right or center based on phone type.

API: PRESS_KEYS(char *keys, int ihold, int ibetween)
Return Value: 0= success, 1 = failure (A value of 1 is returned only if invalid keys are passed), 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char *keys The key, keys or named key that is to be pressed.	A single key (of the form "1"), a number of keys (of the form "123"), or a key(s) to be looked (of the form "<Select>".	Specific parameters not currently defined.
int ihold This is the time in milliseconds to hold each key in the down state.	Integer Values from 1 to 65535	As specified
int ibetween This is the time in milliseconds to delay between the break of one key and the make of the subsequent key.	Integer Values from 1 to 65535	As specified

3.13. RELEASE_KEY

This API will release the specified key.

API: RELEASE_KEY(char *szkey)
Return Value: 0= success, 1 = failure, 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char * szkey The key that is to be released.	Integer Values from 1 to 65535	Integer Values from 1 to 65535

3.14. SELECT_MENU_ITEM

Given a text label, this API will use the up and down keys to find the text label as a highlighted text string and then press the select button of the handset. The API will attempt to navigate down from the current location until it either finds the menu item, reaches a state where the display is not changing with a new keypress (bottom of non-wrapping menu found), or the same menu item as initially selected is again found. If the bottom of the display is reached and the item is not found, the API will return to the top of the menu and search downward until either the item is found or the initially highlighted item is again found. The specific implementation of the navigation shall be determined by the implementer.

API: SELECT_MENU_ITEM(char *szitem)
Return Value: 0= success, 1 = failure (A value of 1 is returned only if the text label cannot be found), 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char *szitem The text label that is to be found.	Varied	Varied

3.15. SET_MODE

This API is used to enable a particular mode. For example, if a mode is boolean, it will determine the current mode and if appropriate select the new mode. If a mode is multi-state, the same process is followed.

API: SET_MODE(char *szMode)
Return Value: 0= success. 1=Failure, no characters sent. 2= Failure, only some characters were sent, 3-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char *szmode A mode that is selectable.	TTY_Enabled	Will cause the TTY mode in the phone to be turned on.
	TTY_Disabled	Will cause the TTY mode in the phone to be turned off.

3.16. SET_OBJECT

This API is used to act upon input objects with defined characteristics. Objects can include strings of text, numeric strings, discretes, audio, etc. The string passed to the API determines the actions associated with the particular object.

API: SET_OBJECT(char *szobject)
Return Value: 0= success, 1 = failure (A value of 1 only if the object cannot be found), 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char *szobject Anything that can be defined as an object that would have simulation characteristics associated with it.	"AnswerCall"	This parameter would cause the system to perform those activities necessary to answer an incoming phone call.
	“SMSSave”	Saves the SMS message to the saved folder on the mobile
	“SK_Ok”	Presses Ok soft key on the mobile
	“SK_View”	Presses View soft key on the mobile
	“SK_Send”	Presses Send soft key on the mobile
	“SK_Ignore”	Presses Ignore soft key on the mobile
	“SK_Read”	Presses Read soft key on the mobile
	“SK_Reply”	Presses Reply soft key on the mobile
	“SK_Back”	Presses Back soft key on the mobile
	“SK_Yes”	Presses Yes soft key on the mobile
	“SK_No”	Presses No soft key on the mobile
	"EndCall"	This parameter would cause the system to perform those activities necessary to terminate a phone call.

3.17. SET_POWER_STATE

This API is used to specify operational/power state in which the DUT should be placed. The API supports both battery connection changes as well as turning the device "on" or "off".

API: SET_POWER_STATE (int ival)

API Variable / Variable Description	**APIAllowable Parameters**	**API Parameter Description and Use**	**Return Value: 0= success, 1 = failure, 4-10 to be defined as needed.**
int ival This variable is used to specify the state of the DUT being requested by the test process and the method for achieving the state.	0 = SP_OFF	Sending this API request to a device using parameter "0" will cause the device to power off (as if a user had initiatied a power off process on the device).	"0" = Success "1"=Failure If the device is already in the on state a return of "0" will be sent.
	1 = SP_ON	Sending this API request to a device using parameter "1" will cause the device to power on (as if a user had initiatied a power on process on the device).	"0" = Success. "1"=Failure.
	2 = SP_BATT_OFF	This parameter will cause the device to disconnect the battery connection from the device under test.	"0" = Success "1"=Failure If the batttery is already connected to the device a return of "0" will be sent.
	3 = SP_BATT_ON	This parameter will cause the device to connect the battery connection to the device under test.	0,1 A value of 1 if object cannot be verified.

3.18. TEXT_FIELD_MUST_BE

This API is used to find the text label, find a text field to the right of it, and then verify that text in that field.

API: TEXT_FIELD_MUST_BE(char *szlabel, char *sztext)
Return Value: 0= success, 1 = failure, 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char *szlabel This is the text label that the API will seach for.	Any valid string	Varied
char *sztext The text that must be present in the labeled field.	Any valid string	Varied

3.19. TEXT_FIELD_MUST_NOT_BE

This API is used to find the text label, find a text field to the right of it, and then verify that the text does not appear in the field.

API: TEXT_FIELD_MUST_NOT_BE(char *szlabel, char *sztext)
Return Value: 0= success, 1 = failure (A value of 1 is returned if the text label cannot be found, or the text in the field matches), 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char *szlabel This is the text label that the API will search for.	Any valid string	Varied
char *sztext The text that must not be present in the labeled field.	Any valid string	Varied

3.20. TEXT_ MUST_BE

The API verifies that specific text is found on the screen.

API: TEXT_MUST_BE(char *sztext)
Return Value: 0= success, 1 = failure (A value of 1 is returned only if the system cannot find the text.), 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char *sztext The text that must be present in the defined search region.	Any valid string	Varied

3.21. TEXT_ MUST_BE_EXT

The API verifies the presence of a specified string within the specified timeout, in the specified region, and with the specified font.

API: TEXT_MUST_BE_EXT(char *sztext, char *fg, char *bg, int x, int y, int endx, int endy, char *font, int itimeout) [2]
Return Value: 0= success, 1 = failure (A value of 1 if text string is not found), 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char *sztext The text that is to be found.	Any valid string	Varied
char *fg Foreground color as a string. E.g. “BLUE”, where color values are defines.	N/A	N/A
char *bg Background color as a string. E.g. “BLUE”, where color values are defines.	N/A	N/A
int x An integer value which is the starting x pixel location of the search region	Integer Values from 1 to 65535	Integer Values from 1 to 65535

[2] Within the specified timeout, in the specified region, with the specified font, verify the presence of the specified string.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
int endy An integer value which is the starting y pixel location of the search region	Integer Values from 1 to 65535	Integer Values from 1 to 65535
int endx An integer value which is the ending x pixel location of the search region	Integer Values from 1 to 65535	Integer Values from 1 to 65535
int y An integer value which is the ending y pixel location of the search region	Integer Values from 1 to 65535	Integer Values from 1 to 65535
char *font Font name from the font table, cannot be the .fnt file name.	Valid font table entry name.	Specific parameters not currently defined.
int itimeout The number of seconds to search for the specified string prior to declaring an error.	-1 to maximum integer value	A negative 1 value (-1) will use a cached image file. Zero is the default timeout value and will perform a single search.

3.22. TEXT_ MUST_NOT_BE

The API verifies the presence of a specified string within the specified timeout, in the specified region, and with the specified font.

API: TEXT_MUST_NOT_BE(char *sztext, int itimeout)
Return Value: 0= success, 1 = failure (A value of 1 is returned only if the system finds the text.),
2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char *sztext The text that must not be present in the defined search region.	Any valid string	Varied
int itimeout The number of seconds to search for the specified string prior to declaring an error.	-1 to maximum integer value.	A negative 1 value (-1) will use a cached image file. Zero is the default timeout value and will perform a single search.

3.23. VERIFY_OBJECT

This API provides a means to verify the specified status of the mobile terminal device. For example, the test may be looking to verify if an incoming call is occurring. The object would specify "incomingCall" and the API would know what that means in the context of the specific mobile terminal (text saying "Incoming Call", Phone ICON, etc.).This API verifies the characteristics associated with anything that can be deemed an object. Object types can include Icons, Softkeys, Images, Tones, LED's, etc.

API: VERIFY_OBJECT(sz szobject)
Return Value: 0= success, 1 = failure (A value of 1 if object cannot be verified.), 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char *szobject A string denoting object.	"CallFailed"	An indicator present when the handset failed to initiate a voice call.
	"Idle"	An indicator present when the handset is idle.
	"InCall"	An indicator present when the handset is within a voice call.
	"IncomingCall"	An indicator present when the handset is receiving notification of an incoming voice call, but has not answered the call.
	"HomeIndicatorOff"	An indicator present when the handset is not in home service area.
	"MainMenu"	An indicator present when the handset is at it's main menu.
	"NoServiceIndicator"	An indicator present when the handset does not detect cellular service.
	"RoamIndicatorFlashing"	An indicator present when the handset is roaming on a non-preferred network.
	"RoamIndicatorOff"	An indicator present when the handset is not roaming.
	"RoamIndicatorOn"	An indicator present when the handset is roaming.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
	"SK_OK"	Verifies if the OK softkey is displayed
	"SK_View"	Verifies if the View softkey is displayed
	"SK_Send"	Verifies if the Send softkey is displayed
	"SK_Ignore"	Verifies if the Ignore softkey is displayed
	"SK_Read"	Verifies if the Read softkey is displayed
	"SK_Reply"	Verifies if the Reply softkey is displayed
	"SK_Back"	Verifies if the Back softkey is displayed
	"SK_Yes"	Verifies if the Yes softkey is displayed
	"SK_No"	Verifies if the No softkey is displayed
	"SMSSendResult"	Verifies if the SMS message was successfully sent
	"SMSNewMessage"	Verifies if the mobile received a new SMS message
	"IC_Pending"	Verifies if Pending message icon is displayed
	"IC_Urgent"	Verifies if Urgent message icon is displayed
	"IC_Cancelled"	Verifies if Cancelled message icon is displayed
	"IC_Delivered"	Verifies if Delivered message icon is displayed
	"CallbackNumber"	Verifies if the CallbackNumber is present in the message body
	"Duplicate"	Verifies if a duplicate SMS message was received
	"OTASound"	Verifies if the OTA session completion message was heard

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
	"DeliveryAck"	Verifies if the mobile received a delivery acknowledgement notification

3.24. WAIT_FOR_TEXT

This API will wait until the text specified either appears within the search area, or the timeout value is reached.

API: WAIT_FOR_TEXT(sz sztext, int itimeout)
Return Value: 0= success (A value of 0 - if the text appears within the specified timeframe), 1 = failure (A value of 1 if timeout is reached), 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
string sztext A string denoting the text that is expected to appear.	Varied	Varied
int itimeout	1 - maximum integer value	This value is the number of milliseconds that will be allowed to elapse until an error is declared. If a value of zero is entered, the test system will immediately perform a single check for the object and declare an error if it does not exist.

3.25. WAIT_FOR_TEXT_TO_NOT_BE

This API will wait until the text specified does not appear within the search area, or the timeout value is reached.

API: WAIT_FOR_TEXT_TO_NOT_BE(sz sztext, int itimeout)
Return Value: 0= success (A value of 0 - if the text disappears or is not found within the specified timeframe), 1 = failure (A value of 1 if timeout is reached, and text remains in the search area) , 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
string sztext A string denoting the text that is expected to appear.	Varied	Varied
int itimeout	1 - maximum integer value	This value is the number of milliseconds that will be allowed to elapse until an error is declared. If a value of zero is entered, the test system will immediately perform a single check for the object and declare an error if it does not exist.

3.26. LOG_PHONE_CONFIGURATION

This API logs information about the phone. The minimum information that should be logged when this API is called is Manufacturer name, handset model, software version, Mobile ID, ESN. If there is other relevant information unique to a particular phone configuration that is available, that information should also be logged. Optional information that may also be logged: TTESN, PRL, ERI, Browser Version, Hardware Version, TTY disabled, and MSID.

API: LOG_PHONE_CONFIGURATION(void)
Return Value: 0= success. This API call always returns a zero.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
void	None	None

3.27. SET_SEARCH_REGION

This API will set the current search region to the coordinates that are specified by the region name passed to the routine. Used when a requirement specifies a particular location.

API: SET_SEARCH_REGION(char *RegionName)
Return Value: 0= success, 1 = failure (A value of 1 is returned only if the search region is not found) , 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char *RegionName The desired search region.	Varied	Varied

3.28. SWITCH_TARGET_TO

This API will change the verification target in multi-target systems. All APIs after this will utilize the current object unless the SWITCH_TARGET_TO is used again to go back to the previous object.

API: SWITCH_TARGET_TO(int itargetname)
Return Value: 0= success, 1 = failure (A value of 1 is returned only if the specified target is undefined in the current test environment) , 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
int itargetname A target type as defined in the test system.	DEFAULT	The target that is defined in the system configuration settings as the default target.
	TARGET_1	The target that is defined in the system configuration settings as the primary target.
	TARGET_2	The target that is defined in the system configuration settings as the secondary target.

3.29. VERIFY_CHOICE_ITEMS

The API verifies that a list of choice items supplied by the call exists on the display. It will pass if the items are visible whether they are selected (reverse highlighted or color coded) or not selected. The API will scroll as necessary to view the complete list. The choice items must be valid screen names.

API: VERIFY_CHOICE_ITEMS(char *alist)
Return Value: 0= success, 1 = failure (A value of 1 is returned only if the specified target is undefined in the current test environment) , 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char *alist This parameter must be a null terminated list of screen names of the following form: char *aList[10] = {"General", "Mobile", "Home", "Work", "Fax", NULL};	Varied	Varied

3.30. VERIFY_CHOICE_ITEMS_SELECTED

This API verifies that a list of choice items supplied by the call exists on the display, and that the items are in the required order and selected (reverse highlighted or color coded) as the cursor passes over each item. This API verifies that the choice items are correct and the initial appropriate choice item is highlighted. The API will scroll as necessary to view the complete list. The choice items must be valid screen names.

API: VERIFY_CHOICE_ITEMS_SELECTED(char *alist)
Return Value: 0= success, 1 = failure (A value of 1 is returned only if the specified target is undefined in the current test environment) , 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
char *alist This parameter must be a null terminated list of screen names of the following form: char *aList[10] = {"General", "Mobile", "Home", "Work", "Fax", NULL};	Varied	Varied

3.31. WAIT_FOR_OBJECT

This API will wait until the object specified either appears in the area specified, or the timeout value is reached.

API: WAIT_FOR_OBJECT(string szobject, int itimeout)
Return Value: 0,1 (A value of 0 - if the object appears within the specified timeframe. A value of 1 if timeout is reached) , 2-10 to be defined as needed.

API Variable / Variable Description	API Allowable Parameters	API Parameter Description and Use
string szobject The object specification is a value obtained from the data file associated with the DUT. The object and be an icon, bitmap, text or other as contained in the spreadsheet. The spreadsheet also contains the allowable search area for the object.	N/A	N/A
int itimeout	Integer Values from 1 to 65535	This value is the number of milliseconds that will be allowed to elapse until an error is declared. If a value of zero is entered, the test system will immediately perform a single check for the object and declare an error if it does not exist.

APPENDIX A: Revision History

Revision	Date	Description of Changes
Rev 1.0	April 2005	• Initial Publication