OMTP PUBLISHED

# OMTP

## MOBILE TERMINAL TESTING

| | |
|---|---|
| **VERSION:** | Version 1.0 |
| **STATUS:** | Approved for Publication |
| **DATE OF LAST EDIT:** | 28th May 2009 |
| **OWNER:** | OMTP Limited |

# CONTENTS

The information contained in this document represents the current view held by OMTP Limited on the issues discussed as of the date of publication.

This document is provided “as is” with no warranties whatsoever including any warranty of merchantability, non-infringement, or fitness for any particular purpose. All liability (including liability for infringement of any property rights) relating to the use of information in this document is disclaimed. No license, express or implied, to any intellectual property rights are granted herein.

This document is distributed for informational purposes only and is subject to change without notice. Readers should not design products based solely on this document.

Each Open Mobile Terminal Platform member and participant has agreed to use reasonable endeavours to inform the Open Mobile Terminal Platform in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. The declared Essential IPR is publicly available to members and participants of the Open Mobile Terminal Platform and may be found on the “OMTP IPR Declarations” list at the OMTP Members Access Area.

The Open Mobile Terminal Platform has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions.

Defined terms and applicable rules above are set forth in the Schedule to the Open Mobile Terminal Platform Member and Participation Annex Form.

# 1 INTRODUCTION

## 1.1 DOCUMENT PURPOSE

### Background

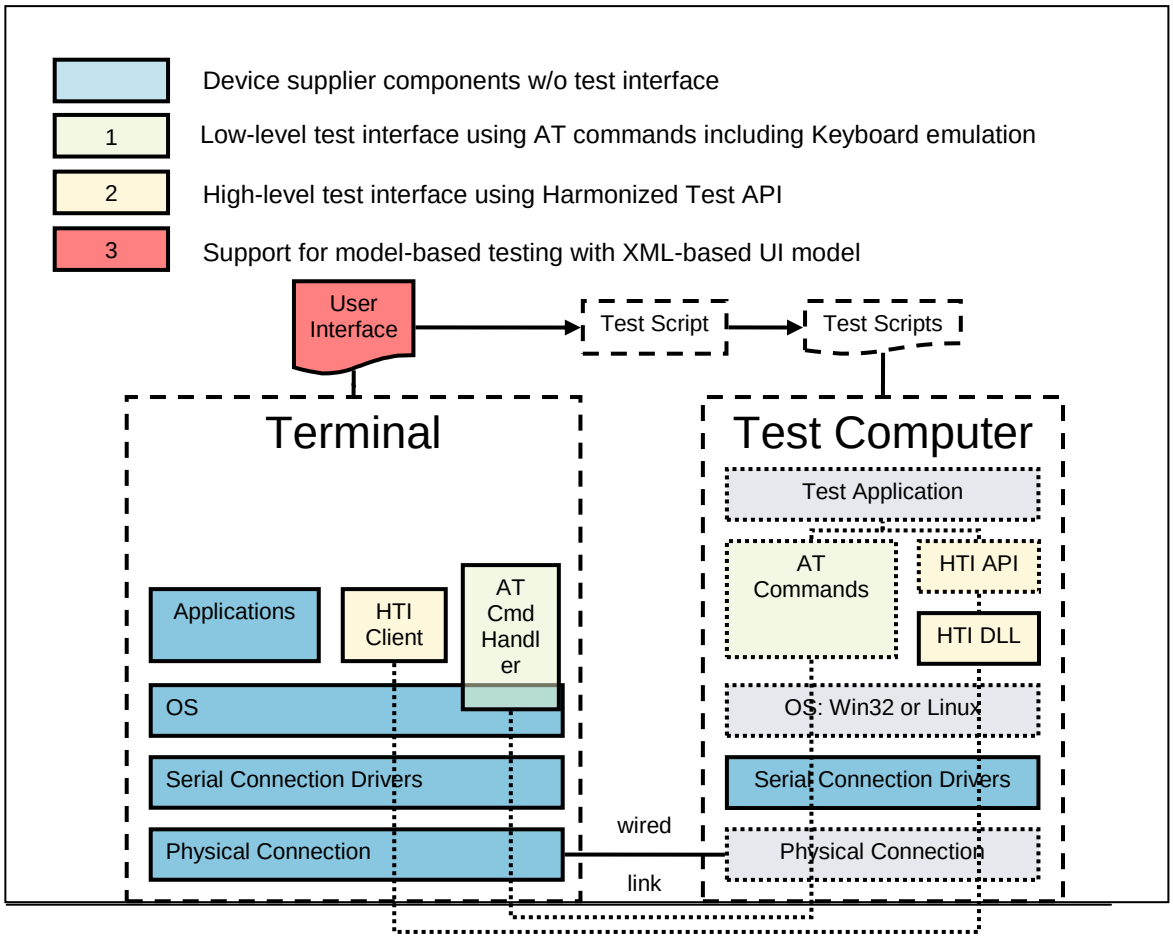
Today's diversity of Terminals and mobile platform implementations results in manual testing across the ecosystem of manufacturers, operators and software developers which in turn drives up costs and time efforts. The market offers testing solutions (for example, Mercury), however all are limited to certain execution environments.

Today, there are no existing standards available that enable consistent automated testing of keyboard input, capturing screenshots, screenshot checksum validation or other Terminal diagnostics such as memory status across platforms.

### Approach

Several methods will be discussed as illustrated below to provide a harmonised interface for Terminal testing.

**Figure 1: Approach for Harmonised Test Interface**



Deliverable

The OMTP Mobile Terminal Testing project will deliver a set of functional Terminal requirements, interfaces and recommendations that are essential for mobile Terminal (Applications and Services) end-to-end test automation using a wired link. Wireless connections, datacards and modems are out of the scope of this document.

## 1.2 BUSINESS RATIONALE

Mobile Terminal testing is an important element in the mobile service delivery value chain. However, the diversity of Terminals and mobile platforms implementations results in manual testing across the ecosystem of manufacturers, operators and software developers. Issues which are faced include:

- The difficulty of replicating Terminal defects reliably
- The inability to accurately perform regression, comparison or benchmark testing between Terminals, warranty period simulation testing, or end-to-end service testing
- The possibility that a defect reaches the end-user ending up with costly returns or poor service take-up
- The cost in resources and time needed for manual testing, with multiple languages and variations

Today the mobile industry faces a fragmented and sometimes absent approach to test automation. This effectively limits the number of Terminals, services and applications that can be launched. Increasing the amount of manual testing that needs to be done before Terminals, services and applications are in service and with the cost and resource constraints. The situation is not likely to change as the number of Terminals, variations and the pace of development is constantly increasing.

Having a common set of APIs implemented across different execution environments enable automated testing. However, controlled and secure access would greatly limit manual testing efforts for both manufacturers and operators.

## 1.3 INTENDED AUDIENCE

The audiences for this recommendation:

- Any party conducting functional testing with mobile Terminals including Operators and Terminal vendors.

- Standards organisations taking the OMTP output as input for deeper standardisation to facilitate automated and secure testing across execution environments.
- The recommendation will be available for request by operators and delivered by Terminal manufacturers, platform and application vendors. The final consistent testing interfaces can then be used by all those parties (vendors, operators, GCF, OMSI, etc.) conducting Terminal testing.

## 1.4 CONVENTIONS

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC2119 [1].

- MUST: This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
- MUST NOT: This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.
- SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.
- MAY: This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

The requirements within this document are uniquely identified using the following format:

MTT-####(.#.#)(a), where:

- MTT is the acronym identifying the subject of this OMTP document (Mobile Terminal Testing)
- #### is a 4 digit number that identifies the requirement (e.g. 0020) and which is to be unique within the document.
- (.#.#) are numbers that indicate sub-requirements (e.g. 00020.1 & 00020.2 which would be sub-requirements of 00020 and 00020.1.1 & 00020.1.2 which would be sub-requirements of 00020.1)
- (a) is one lower-case letter used to indicate a minor revision has been made to the requirement definition.

# 2 USE CASES

## 2.1 KEY OPERATOR USE CASES

The key operator use cases are listed below in priority order.

### 2.1.1 USE CASE GROUPING 1 – AUTOMATED TESTING PRESALES

Automated testing of Terminals before market launch via a test application resident on a test computer.

The possible testing scope might cover:

- Application testing to verify the correct functionality of e.g. browser, MMS, E-Mail,
- Stress and reliability testing,
- Detection of memory leaks,
- Performance measurements,
- Analysis of influence of external parameters (e.g. Network) through good reproducible test scenarios.

### 2.1.2 USE CASE GROUPING 2 – AUTOMATED TESTING POST-SALES

Automated testing of Terminals post-sales (e.g. in-store customer service scenarios) via a test application resident on a test computer, for example by re-using the same interfaces that are used for diagnostics purposes.

### 2.1.3 USE CASE GROUPING 3 – NETWORK MONITORING

Automated monitoring of network functionality by periodical usage and an automated analysis of network functionality, performance and stability should be enabled.

## 2.2 POSSIBLE FUTURE OPERATOR USE CASES

The following use cases are out of scope of this document. They may however be addressed in a future release of this document.

### 2.2.1 USE CASE GROUPING 1 – REMOTE TESTING

Managing test sessions via a testing application installed on the Terminal.

### 2.2.2 USE CASE GROUPING 2 – DISABLED PEOPLE

An additional benefit of test control will be the ability of easier control of the Terminal for disabled people. The Terminal can be connected to a special computer for handicapped people which allows an external

control for the Terminal, for example using a Braille reader or large zoom in case of refractive errors.

# 3 GENERAL REQUIREMENTS

To enable end to end automated Terminal testing, this section outlines and groups the underlying requirements as follows:

- General Terminal Requirements
- General Attention (AT) Commands
- Extension of AT Commands
- Abstracted Command Interface
- General Connectivity Requirements
- Security Requirements
- Navigation Aid Requirements

## *3.1 GENERAL TERMINAL REQUIREMENTS*

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-0010** | The functionality and system performance of the Terminal MUST be identical if accessed through the test control versus if a Terminal button is manually pressed, e.g. no major additional delay by the Application Execution Environment (AEE) in processing the command. |
| **MTT-0020** | The Terminal SHOULD support the test access to general Terminal information that is also available to the user. |
| **MTT-0030** | The Terminal MAY support the tracing of networking IP traffic. The output file should be in a standardised format, e.g. LIBCAP (only broadcasts and information directly addressed to the Terminals interface SHALL be capture. Capturing in promiscuous mode SHALL not be allowed). |
| **MTT-0040** | The device SHOULD support the capture of the display as a screenshot in a defined format (e.g. bitmap) such that it can be re-used for further diagnostics. |
| **MTT-0040.1** | The performance of the Terminal SHOULD not be influenced significantly when reading out the display. |
| **MTT-0050** | The Terminal MUST be rechargeable when connected to a test computer using the cable connection (e.g. USB) between the Terminal and the Test Computer or using a parallel charger connectivity. |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-0050.1** | The chargers MUST provide enough current to supply the Terminal under normal testing circumstances including all commands that are specified in this document. |
| **MTT-0060** | The Terminal SHOULD provide programmatic access to every physical key on the Terminal. |
| **MTT-0070** | The Terminal MUST support either the AT command Interface (as described in chapter 3.2 and 3.3) or the HTI interface (as described in chapter 3.4). |
| **MTT-0070.1** | If the Terminal only supports HTI, the manufacturer SHOULD deliver a possibility to control the radio stack through the test control. |
| **MTT-0080** | If the device supports HTI, the device manufacturer SHALL enable access by network operators, test houses or other authorised agencies to the HTI using a secure access mechanism which SHALL prevent unintentional, accidental and limit the threat of malicious access.<br>Note: The access method used shall be the choice of the individual device manufacturer and may include, but shall not be restricted to, specialised hardware dongles, hardware adapters, activation/deactivation using authorised key stroke entry, activation using a digitally signed client software installation. |
| **MTT-0080.1** | There MUST be a physical interaction of the user on the Terminal under test before the HTI can be utilised. |
| **MTT-0080.2** | If the Terminal supports HTI, the access to the HTI MUST be possible with each (commercial) software. |
| **MTT-0080.3** | If the HTI access is controlled by software there SHOULD be an initial user confirmation when enabling the HTI. |
| **MTT-0080.4** | The access to the HTI must not be affected by power-cycling the phone. |
| **MTT-0090** | If the Terminal supports AT command interfaces and HTI interface, it SHOULD be possible to use both in parallel. |

## 3.2 General AT Commands

This section covers AT commands already specified in 3GPP [3] and [4], especially those commands which are essential to enable Terminal testing through an AT command interface, but defined as optional.

| Req. Id | Requirement |
|---|---|
| | ITU-T V.250 / 3GPP TS 27.007 Basic AT commands |
| **MTT-0100** | The Terminal MUST support the &F or Z command ([3], Chapter 5.8). |
| **MTT-0110** | The Terminal MUST support the I command ([3], Chapter 5.8). |
| **MTT-0120** | The Terminal MUST support the O command ([3], Chapter 6.25). |
| **MTT-0130** | The Terminal MUST support the A command ([3], Chapter 6.25). |
| **MTT-0140** | The Terminal MUST support the D command ([3], Chapter 6.25). |
| **MTT-0150** | The Terminal MUST support the H command ([3], Chapter 6.25). |
| **MTT-0160** | The Terminal MUST support the E command ([3], Chapter 6.2.4). |
| **MTT-0170** | The Terminal MUST support the S0 command ([3], Chapter 6.25). |
| **MTT-0180** | The Terminal MUST support the FCLASS command ([3], Annex C.2.1). |
| | 3GPP TS 27.007 |
| **MTT-0190** | The Terminal MUST support the +CGMI command ([3], Chapter 5.1). |
| **MTT-0200** | The Terminal MUST support the +CGMM command ([3], Chapter 5.2). |
| **MTT-0210** | The Terminal MUST support the +CGMR command ([3], Chapter 5.3). |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-0220** | The Terminal MUST support the +CGSN command ([3], Chapter 5.4). |
| **MTT-0230** | The Terminal MUST support the +CIMI command ([3], Chapter 5.6). |
| **MTT-0240** | The Terminal MUST support the +CHUP command ([3], Chapter 6.5). |
| **MTT-0250** | The Terminal MUST support the +CBST command ([3], Chapter 6.7). |
| **MTT-0260** | The Terminal MAY support the +CNUM command ([3], Chapter 7.1). |
| **MTT-0270** | The Terminal MUST support the +COPS command ([3], Chapter 7.3). |
| **MTT-0280** | The Terminal SHOULD support the +CLCC command ([3], Chapter 7.18). |
| **MTT-0290** | The Terminal MAY support the +COPN command ([3], Chapter 7.21). |
| **MTT-0300** | The Terminal MUST support the +CPAS command ([3], Chapter 8.1). |
| **MTT-0310** | The Terminal MUST support the +CBC command ([3], Chapter 8.4). |
| **MTT-0320** | The Terminal MUST support the +CSQ command ([3], Chapter 8.5). |
| **MTT-0330** | The Terminal MUST support the +CMEC command ([3], Chapter 8.6). |
| **MTT-0330.1** | The Terminal MUST support the keypad options 0,1 & 2 in the CMEC command. |
| **MTT-0330.2** | The Terminal MUST support the display options 0,1 & 2 in the CMEC command. |
| **MTT-0330.3** | The Terminal MUST support the indication options 0,1 & 2 in the CMEC command. |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-0330.4** | A Terminal controlled by a touchscreen MUST support the touch screen option 0,1 & 2 as specified in [3]. |
| **MTT-0330.5** | The Terminal MUST support the touch indication options 0,1 & 2 in the CMEC command if the Terminal has touch-screen functionality. |
| **MTT-0340** | The Terminal MUST support the +CKPD command ([3], Chapter 8.7). |
| **MTT-0340.1** | The Terminal SHOULD support all physical keys mapping when using the CKPD command. |
| **MTT-0340.2** | The Terminal MUST support the ‘time’ attribute in the CKPD command. |
| **MTT-0340.3** | The Terminal SHOULD support all proprietary keys using the ‘:’ option in the CKPD command. |
| **MTT-0340.4** | The Terminal SHOULD support the ‘;’ option (string entering) in the CKPD command. |
| **MTT-0350** | The Terminal MAY support the +CDIS command ([3], Chapter 8.8). |
| **MTT-0350.1** | The Terminal SHOULD support the ‘CDIS?’ option in the CDIS command. |
| **MTT-0360** | The Terminal MUST support the +CIND command ([3], Chapter 8.9). |
| **MTT-0360.1** | The Terminal SHOULD support the input status indication as specified in [3]. |
| **MTT-0370** | The Terminal MUST support the +CMER command ([3], Chapter 8.10). |
| **MTT-0370.1** | The Terminal SHOULD support keyboard and event reporting in the CMER command. |
| **MTT-0370.2** | The Terminal MAY support display event reporting in the CMER command. |
| **MTT-0370.3** | The Terminal SHOULD support indication event reporting in the CMER command. |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-0370.4** | The Terminal SHOULD support touch screen indication reporting in the CMER command [3]. |
| **MTT-0370.5** | The Terminal SHOULD support the CKEV event when the keyboard setting is 1 or 2 in the CMER command. |
| **MTT-0370.6** | The Terminal MAY support the CDEV event when the display setting is 1 or 2 in the CMER command. |
| **MTT-0370.7** | The Terminal SHOULD support the CIEV event when the indication setting is 1 or 2 in the CMER command. |
| **MTT-0370.8** | The Terminal SHOULD support the CTEV event when touch screen setting is 1 or 2 in the CMER command [3]. |
| **MTT-0380** | The Terminal MUST support the +CPBS command ([3], Chapter 8.11). |
| **MTT-0390** | The Terminal SHOULD support the +CPBR command ([3], Chapter 8.12). |
| **MTT-0400** | The Terminal SHOULD support the +CPBW command ([3], Chapter 8.13). |
| **MTT-0410** | The Terminal SHOULD support the +CCLK command ([3], Chapter 8.9). |
| **MTT-0420** | The Terminal SHOULD support the +CLAN command ([3], Chapter 8.30). |
| **MTT-0420.1** | The Terminal SHOULD support all installed languages through the +CLAN command. |
| **MTT-0420.2** | The supported languages MAY exceed the list specified in the Terminal's documentation. |
| **MTT-0420.3** | The Terminal SHOULD support the +CLAE event ([3], Chapter 8.31). |
| **MTT-0430** | The Terminal MUST support the +CMAR command ([3], Chapter 8.36). |
| **MTT-0440** | The Terminal SHOULD support the +CLAC command ([3], Chapter 8.37). |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-0450** | The Terminal MUST support the +CMEE command ([3], Chapter 9.1). |
| **MTT-0450.1** | The Terminal SHOULD support the +CME event. |
| **MTT-0450.2** | The Terminal SHOULD support option 0,1 and 2 in the CMEE command. |
| **MTT-0460** | The Terminal MUST support the +CBKLT command ([3], Chapter 8.51). |
| **MTT-0470** | The Terminal MUST support the +CSCS command ([3], Chapter 5.5). |
| **MTT-0480** | The Terminal MUST support the +CLCK command ([3], Chapter 7.4)**.** |
| **MTT-0490** | The Terminal SHOULD support the +CPWD command ([3], Chapter 7.5). |
| **MTT-0500** | The Terminal MUST support the +CREG command ([3], Chapter 7.2). |
| **MTT-0510** | The Terminal MUST support the +CPOL command ([3], Chapter 7.19). |
| **MTT-0520** | The Terminal MUST support the +CFUN command ([3], Chapter 8.2). |
| **MTT-0530** | The Terminal MAY support the +CPIN command ([3], Chapter 8.3). |
| **MTT-0530.1** | The +CPIN support is conditional assuming all password entry can be performed using the mandatory +CKPD command. |
| **MTT-0540** | The Terminal SHOULD support the +CPBF command ([3], Chapter 8.13). |
| **MTT-0550** | The Terminal MUST support the +CGDCONT command ([3], Chapter 10.1.1). |
| **MTT-0560** | The Terminal MUST support the +CGDSCONT command ([3], Chapter 10.1.2). |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-0570** | The Terminal MUST support the +CGTFT command ([3], Chapter 10.1.3). |
| **MTT-0580** | The Terminal SHOULD support the +CGQREQ command ([3], Chapter 10.1.4). |
| **MTT-0590** | The Terminal SHOULD support the +CGQMIN command ([3], Chapter 10.1.5). |
| **MTT-0600** | The Terminal MUST support the +CGEQREQ command ([3], Chapter 10.1.6). |
| **MTT-0610** | The Terminal MUST support the +CGEQNEG command ([3], Chapter 10.1.8). |
| **MTT-0620** | The Terminal MUST support the +CGATT command ([3], Chapter 10.1.9). |
| **MTT-0630** | The Terminal MUST support the +CGCMOD command ([3], Chapter 10.1.11). |
| **MTT-0640** | The Terminal MUST support the +CGDATA command ([3], Chapter 10.1.12). |
| **MTT-0650** | The Terminal MUST support the +CGPADDR command ([3], Chapter 10.1.14). |
| **MTT-0660** | The Terminal SHOULD support the +CGCLASS command ([3], Chapter 10.1.17). |
| **MTT-0670** | The Terminal MUST support the +CGEREP command ([3], Chapter 10.1.18). |
| **MTT-0680** | The Terminal SHALL support the +CGREG command ([3], Chapter 10.1.19). |
| **MTT-0690** | The Terminal SHALL support the +CGSMS command ([3], Chapter 10.1.20). |
| **MTT-0700** | The Terminal MUST support the +CSO command ([3], Chapter 8.53). |
| **MTT-0710** | The Terminal MUST support the +CSS command ([3], Chapter 8.54). |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-0720** | The Terminal SHOULD support the +CTSA command ([3], Chapter 8.52). |
| **MTT-0730** | The Terminal MUST support the +CEER command ([3], Chapter 6.10). |
| **MTT-0740** | The Terminal MUST support the +CVHU command ([3], Chapter 6.20). |
| **MTT-0750** | The Terminal MUST support the +CLIP command ([3], Chapter 7.6). |
| **MTT-0760** | The Terminal MUST support the +CCWA command ([3], Chapter 7.12). |
| **MTT-0770** | The Terminal MUST support the +CHLD command ([3], Chapter 7.13). |
| **MTT-0780** | The Terminal SHOULD support the +CAOC command ([3], Chapter 7.16). |
| **MTT-0790** | The Terminal MUST support the +CLVL command ([3], Chapter 7.23). |
| **MTT-0800** | The Terminal MUST support the +CMUT command ([3], Chapter 7.24). |
| **MTT-0810** | The Terminal SHOULD support the +CACM command ([3], Chapter 7.25). |
| **MTT-0820** | The Terminal SHOULD support the +CAMM command ([3], Chapter 7.26). |
| **MTT-0830** | The Terminal MUST support the +CGEQMIN command ([3], Chapter 10.1.7). |
| **MTT-0840** | The Terminal MUST support the +CGACT command ([3], Chapter 10.1.10). |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-0850** | The Terminal MUST support the +CR command ([3], Chapter 6.9). |
| **MTT-0860** | The Terminal MUST support the +CRC command ([3], Chapter 6.11). |
| **MTT-0870** | The Terminal MUST support the +CRLP command ([3], Chapter 6.6.8). |
| **MTT-0880** | The Terminal MUST support the +CMOD command ([3], Chapter 6.4) if alternating mode calls are implemented. |
| **MTT-0890** | The Terminal SHOULD support either the +CRSM command or the +CSIM command ([3], Chapter 8.17). |
| | 3GPP TS 27.005 – Messaging AT commands |
| **MTT-0900** | The Terminal MUST support the +CMGF command ([4], Chapter 3.2.3). |
| **MTT-0910** | The Terminal MUST support the +CSCA command ([4], Chapter 3.3.1). |
| **MTT-0920** | The Terminal MUST support the +CNMI command ([4], Chapter 3.4.1). |
| **MTT-0930** | The Terminal MUST support the +CMGD command ([4], Chapter 3.5.4). |
| **MTT-0940** | The Terminal MUST support the +CMGL command ([4], Chapter 3.4.2). |
| **MTT-0950** | The Terminal MUST support the +CMGR command ([4], Chapter 3.4.3). |
| **MTT-0960** | The Terminal MUST support the +CMGS command ([4], Chapter 3.5.1). |
| **MTT-0970** | The Terminal MUST support the +CPMS command ([4], Chapter 3.2.2). |
| **MTT-0980** | The Terminal MUST support the +CMS event errors ([4], Chapter 3.2.5). |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-0990** | The Terminal SHOULD support the +CSCB command ([4], Chapter 3.3.4). |
| **MTT-1000** | The Terminal MUST support the +CMSS command ([4], Chapter 3.5.2). |
| **MTT-1010** | The Terminal MUST support the +CMGD command ([4], Chapter 3.5.3). |
| **MTT-1020** | The Terminal MUST support the +CMGC command ([4], Chapter 3.5.5). |
| **MTT-1030** | The Terminal MUST support the +CSMS command ([4], Chapter 3.2.1). |

## 3.3 Extension of AT Commands

This chapter covers necessary extensions to the AT command set which are required to enable Terminal testing in a way it is not yet supported through the already specified AT commands (where applicable).

| Req. Id | Requirement |
|---|---|
| **MTT-1100** | The Terminal SHOULD support the CRST command.<br>***[Reset control].*** |
| **MTT-1110** | The Terminal SHOULD support the extended CBC command.<br>***[Charge battery control].*** |
| **MTT-1120** | The Terminal SHOULD support the CTXPWR command.<br>***[UE transmitted power].*** |
| **MTT-1130** | The Terminal SHOULD support the CWCDMASQ command.<br>***[WCDMA Signal Quality].*** |

Format of the above commands can be found in this document appendix (Chapter 9).

## 3.4 ABSTRACTED COMMAND INTERFACE

This section includes requirements for an abstracted test interface either based on AT commands or proprietary solutions such as PC or Terminal clients to simplify and harmonise the execution of more complex test scenarios.

| REQ. ID | REQUIREMENT |
|---|---|
| | Requirements refer to section 4.4.1 HTI System Service |
| **MTT-1200** | The Terminal SHOULD support the 'Authentication' command as specified in Section 4.4.1.1. |
| **MTT-1200.1** | The Terminal MAY support different authentication levels to restrict some HTI commands. |
| **MTT-1210** | The Terminal MUST support the 'HTI Version' command as specified in Section 4.4.1.2. |
| **MTT-1220** | The Terminal MUST support the 'Services List' command as specified in Section 4.4.1.3. |
| **MTT-1230** | The Terminal MUST support the 'Stop' command as specified in Section 4.4.1.4. |
| **MTT-1240** | The Terminal MUST support the 'Reboot' command as specified in Section 4.4.1.5. |
| **MTT-1250** | The Terminal MUST support the 'Reset' command as specified in Section 4.4.1.7. |
| **MTT-1260** | The Terminal MUST support the 'Restore Factory Settings' command as specified in Section 4.4.1.6. |
| **MTT-1270** | The Terminal SHOULD support the 'Show Console' command as specified in Section 4.4.1.8. |
| **MTT-1280** | The Terminal SHOULD support the 'Hide Console' command as specified in Section 4.4.1.9. |
| **MTT-1290** | The Terminal SHOULD support the 'Instance ID' command as specified in Section 4.4.1.10. |
| **MTT-1300** | The Terminal MAY support the 'Debug Print' command as specified in Section 4.4.1.11. |
| | Requirements refer to section 4.4.2 HTI Echo Service |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-1310** | The Terminal MUST support the Echo Service as specified in Section 4.4.2. |
| | Requirements refer to section 4.4.3 HTI Key Event Service |
| **MTT-1315** | The manufacturer MUST provide a list of the supported scancodes for the Key Event Service. |
| **MTT-1320** | The Terminal MUST support the 'Single Key press' command as specified in Section 4.4.3.1. |
| **MTT-1330** | The Terminal MUST support the 'Key down' command as specified in Section 4.4.3.2. |
| **MTT-1340** | The Terminal MUST support the 'Key up' command as specified in Section 4.4.3.3. |
| **MTT-1350** | The Terminal MUST support the 'Type Text' command as specified in Section 4.4.3.4. |
| **MTT-1350.1** | The 'Type Text' command MUST not be affected by predictive text entry mechanisms e.g. T9, I-Tap, Zi Corp, Suretype. |
| **MTT-1350.2** | The 'Type Text' command MUST work on all supported Terminal languages. |
| **MTT-1360** | The Terminal MUST support the 'Long Key press' command as specified in Section 4.4.3.5. |
| **MTT-1370** | The Terminal MAY support the 'Type Text Password' command as specified in Section 4.4.3.6. |
| **MTT-1380** | The Terminal MUST support the 'Key Press Sequence' command as specified in Section 4.4.3.7. |
| **MTT-1390** | If the Terminal supports touchscreen, it MUST support the 'Tap Screen' command as specified in Section 4.4.3.8. |
| **MTT-1400** | If the Terminal supports touchscreen, it MUST support the 'Tap and Drag' command as specified in Section 4.4.3.9. |
| **MTT-1410** | If the Terminal supports touchscreen and is capable of handling multipoint events, it MUST support the 'Tap and drag multipoint' command as specified in Section 4.4.3.10. |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-1420** | If the Terminal supports touchscreen, it MUST support the 'Pointer down' command as specified in Section 4.4.3.11. |
| **MTT-1430** | If the Terminal supports touchscreen, it MUST support the 'Pointer up' command as specified in Section 4.4.3.12. |
| | Requirements refer to section 4.4.4 HTI Screenshot Service |
| **MTT-1440** | The Terminal MUST support the 'Full Screen capture' command as specified in Section 4.4.4.1.<br>Note: Restrictions on the usage if this command MAY apply due to limited performance of the Terminal. |
| **MTT-1440.1** | The Terminal MUST support the Bitmap file format as output for Screen capture. |
| **MTT-1440.2** | The Terminal SHOULD support the JPG, PNG, GIF file format as output for screen capture. |
| **MTT-1440.3** | The Terminal MAY support any other standardised image or video file formats as output for screen capture. |
| **MTT-1450** | The Terminal SHOULD support the 'Screen Region capture' command as specified in Section 4.4.4.2. |
| **MTT-1460** | The Terminal SHOULD support the 'Text Recognition' command as specified in Section 4.4.4.3. |
| **MTT-1470** | The Terminal MAY support the 'Text Bitmap' command as specified in Section 4.4.4.4. |
| **MTT-1480** | The Terminal MUST support the 'Full Screen capture in Series' command as specified in Section 4.4.4.5. |
| **MTT-1490** | The Terminal MUST support the 'Screen Region capture in Series' command as specified in Section 4.4.4.6. |
| **MTT-1500** | The Terminal SHOULD support the 'Full Screen Delta capture' command as specified in Section 4.4.4.9. |
| **MTT-1510** | The Terminal SHOULD support the 'Screen Region Delta capture' command as specified in Section 4.4.4.10. |
| **MTT-1520** | The Terminal SHOULD support the 'Screen Delta reset' command as specified in Section 4.4.4.11. |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-1530** | The Terminal MUST support the 'Select Screen' command as specified in Section 4.4.4.7. |
| **MTT-1540** | The Terminal MUST support the 'Screen Mode' command as specified in Section 4.4.4.8. |
| | Requirements refer to section 4.4.5 HTI Application Control Service |
| **MTT-1550** | The Terminal SHOULD support the 'Start Process' command as specified in Section 4.4.5.1. |
| **MTT-1560** | The Terminal SHOULD support the 'Stop Process' command as specified in Section 4.4.5.2. |
| **MTT-1570** | The Terminal SHOULD support the 'Stop Process by ID' command as specified in Section 4.4.5.3. |
| **MTT-1580** | The Terminal SHOULD support the 'Get Process Status' command as specified in Section 4.4.5.4. |
| **MTT-1590** | The Terminal SHOULD support the 'Get Process Status by ID' command as specified in Section 4.4.5.5. |
| **MTT-1600** | The Terminal MUST support the 'Start Application by name' command as specified in Section 4.4.5.6. |
| **MTT-1600.1** | The 'Start Application by name' command MUST be supported on embedded Java Runtime Environments. |
| **MTT-1610** | The Terminal MUST support the 'Start Application by UID' command as specified in Section 4.4.5.7 if applicable on the Terminal. |
| **MTT-1620** | The Terminal SHOULD support the 'Start Document' command as specified in Section 4.4.5.8. |
| **MTT-1630** | The Terminal MUST support the 'Get Application Status by application name' command as specified in Section 4.4.5.9. |
| **MTT-1640** | The Terminal SHOULD support the 'Get Application Status by Document name' command as specified in Section 4.4.5.10. |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-1650** | The Terminal MUST support the 'Get Application Status by UID' command as specified in Section 4.4.5.11 if applicable on the Terminal. |
| **MTT-1660** | The Terminal MUST support the 'Stop Application by application name' command as specified in Section 4.4.5.12. |
| **MTT-1670** | The Terminal MUST support the 'Stop Application by UID' command as specified in Section 4.4.5.14 if applicable on the Terminal. |
| **MTT-1680** | The Terminal SHOULD support the 'Stop Application by Document name' command as specified in Section 4.4.5.13. |
| **MTT-1690** | The Terminal MUST support the 'List Running Applications' command as specified in Section 4.4.5.15. |
| **MTT-1700** | The Terminal SHOULD support the 'List Processes' command as specified in Section 4.4.5.16. |
| **MTT-1710** | The Terminal MAY support the 'Start Process for Exit Code' command as specified in Section 4.4.5.17. |
| **MTT-1720** | The Terminal MAY support the 'Get Process Exit Code' command as specified in Section 4.4.5.18. |
| **MTT-1730** | The Terminal MUST support the 'Install Software' command as specified in Section 4.4.5.19. |
| **MTT-1740** | The Terminal MUST support the 'Uninstall Software' command as specified in Section 4.4.5.20. |
| **MTT-1750** | The Terminal MUST support the 'Uninstall Software by Name' command as specified in Section 4.4.5.21. |
| **MTT-1750.1** | The 'Install Software', 'Uninstall Software' and 'Uninstall Software by Name' SHOULD apply on applications for native operating system. |
| **MTT-1750.2** | The 'Install Software', 'Uninstall Software' and 'Uninstall Software by Name' SHOULD apply on applications in high level language, e.g. JAVA MIDP. |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-1750.3** | The 'Install Software', 'Uninstall Software' and 'Uninstall Software by Name' MUST NOT affect preinstalled applications and the operating system. |
| | Requirements refer to section 4.4.6 HTI FTP Service |
| **MTT-1760** | The Terminal MUST support the 'Upload File' command as specified in Section 4.4.6.1. |
| **MTT-1770** | The Terminal MUST support the 'Download File' command as specified in Section 4.4.6.2. |
| **MTT-1780** | The Terminal MUST support the 'cancel File Transfer' command as specified in Section 4.4.6.3. |
| **MTT-1790** | The Terminal MUST support the 'List Files' command as specified in Section 4.4.6.4. |
| **MTT-1800** | The Terminal SHOULD support the 'List Files with Sizes' command as specified in Section 4.4.6.6. |
| **MTT-1800.1** | The Terminal SHOULD support display all file attributes which are supported by the Terminal when executing the 'List Files with Sizes'. |
| **MTT-1810** | The Terminal MUST support the 'List Directories' command as specified in Section 4.4.6.5. |
| **MTT-1820** | The Terminal SHOULD support the 'Create Directory' command as specified in Section 4.4.6.7. |
| **MTT-1830** | The Terminal SHOULD support the 'Delete Directory' command as specified in Section 4.4.6.8. |
| **MTT-1840** | The Terminal MUST support the 'Delete File' command as specified in Section 4.4.6.9. |
| **MTT-1850** | The Terminal MAY support the 'Set Forced Operations' command as specified in Section 4.4.6.10. |
| **MTT-1860** | The Terminal MUST support the 'File Checksum' command as specified in Section 4.4.6.11. |
| **MTT-1870** | The Terminal MAY support the 'Format' command as specified in Section 4.4.6.12. |

| REQ. ID | REQUIREMENT |
|---|---|
| | Requirements refer to section 4.4.7 HTI Sys Info Service |
| **MTT-1880** | The Terminal SHOULD support the 'HAL' command as specified in Section 4.4.7.1. |
| **MTT-1890** | The Terminal MUST support the 'IMEI' command as specified in Section 4.4.7.2. |
| **MTT-1890.1** | For non-GSM Terminals the hardware specific serial number MUST returned when using the 'getImei' command e.g. ESN. |
| **MTT-1900** | The Terminal MUST support the 'SW Version' command as specified in Section 4.4.7.3. |
| **MTT-1910** | The Terminal MUST support the 'LangVersion' command as specified in Section 4.4.7.4. |
| **MTT-1920** | The Terminal MUST support the 'SWLangVersion' command as specified in Section 4.4.7.5. |
| **MTT-1930** | The Terminal MUST support the 'UserAgentString' command as specified in Section 4.4.7.6. |
| **MTT-1930.1** | The 'getUserAgentString' MUST return the UA string of the default browser. |
| **MTT-1940** | The Terminal MUST support the 'SetHomeTime' command as specified in Section 4.4.7.17. |
| **MTT-1950** | The Terminal MUST support the 'GetHomeTime' command as specified in Section 4.4.7.18. |
| **MTT-1960** | The Terminal SHOULD support the 'DateTimeFormat' command as specified in Section 4.4.7.19. |
| **MTT-1970** | The Terminal MUST support the 'GetFreeRAM' command as specified in Section 4.4.7.7. |
| **MTT-1980** | The Terminal MUST support the 'GetUsedRAM' command as specified in Section 4.4.7.8. |
| **MTT-1990** | The Terminal MUST support the 'GetTotalRAM' command as specified in Section 4.4.7.9. |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-2000** | The Terminal MAY support the ‘EatRAM’ command as specified in Section 4.4.7.10. |
| **MTT-2010** | The Terminal MAY support the ‘ReleaseRAM’ command as specified in Section 4.4.7.11. |
| **MTT-2020** | The Terminal MUST support the ‘GetFreeDiskSpace’ command as specified in Section 4.4.7.12. |
| **MTT-2030** | The Terminal MUST support the ‘GetUsedDiskSoace’ command as specified in Section 4.4.7.13. |
| **MTT-2040** | The Terminal MUST support the ‘GetTotalDiskSpace’ command as specified in Section 4.4.7.14. |
| **MTT-2050** | The Terminal MAY support the ‘EatDiskSpace’ command as specified in Section 4.4.7.15. |
| **MTT-2060** | The Terminal MAY support the ‘ReleaseDiskSpace’ command as specified in Section 4.4.7.16. |
| **MTT-2070** | The Terminal MUST support the ‘LightStatus’ command as specified in Section 4.4.7.21. |
| **MTT-2080** | The Terminal MUST support the ‘LightOn’ command as specified in Section 4.4.7.22. |
| **MTT-2090** | The Terminal MUST support the ‘LightOff’ command as specified in Section 4.4.7.23. |
| **MTT-2100** | The Terminal SHOULD support the ‘LightBlink’ command as specified in Section 4.4.7.24. |
| **MTT-2110** | The Terminal MUST support the ‘LightRelease’ command as specified in Section 4.4.7.25. |
| **MTT-2120** | The Terminal MUST support the ‘ScreenSaverDisable’ command as specified in Section 4.4.7.26. |
| **MTT-2130** | The Terminal MUST support the ‘ScreenSaverEnable’ command as specified in Section 4.4.7.27. |
| **MTT-2140** | The Terminal MUST support the ‘ScreenSaver Timeout’ command as specified in Section 4.4.7.28. |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-2150** | The Terminal MUST support the ‘GetNetworkMode’ command as specified in Section 4.4.7.29. |
| **MTT-2160** | The Terminal MUST support the ‘SetNetworkMode’ command as specified in Section 4.4.7.30. |
| **MTT-2170** | The Terminal MUST support the ‘SetNetworkModeNoReboot’ command as specified in Section 4.4.7.31. |
| **MTT-2180** | The Terminal SHOULD support the ‘setHsdpa’ command as specified in Section 4.4.7.32. |
| **MTT-2190** | The Terminal MAY support the ‘IrActivate’ command as specified in Section 4.4.7.33. |
| **MTT-2200** | The Terminal MUST support the ‘BtPower’ command as specified in Section 4.4.7.34. |
| **MTT-2210** | The Terminal MUST support the ‘BtSettings’ command as specified in Section 4.4.7.35. |
| **MTT-2220** | The Terminal MUST support the ‘BtDeletePairings’ command as specified in Section 4.4.7.36. |
| **MTT-2230** | The Terminal SHOULD support the ‘KeyLockToggle’ command as specified in Section 4.4.7.37. |
| **MTT-2240** | The Terminal MAY support the ‘AutoKeyLockTime’ command as specified in Section 4.4.7.38. |
| **MTT-2250** | The Terminal MUST support the ‘EmptyDrmRightsDb’ command as specified in Section 4.4.7.39. |
| **MTT-2260** | The Terminal MUST support the ‘BatteryStatus’ command as specified in Section 4.4.7.40. |
| **MTT-2270** | The Terminal MUST support the ‘SignalStrength’ command as specified in Section 4.4.7.41. |
| **MTT-2280** | The Terminal SHOULD support the ‘UpdateMediaGallery’ command as specified in Section 4.4.7.42. |
| | Requirements refer to section 4.4.8 HTI Audio Control Service |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-2290** | The Terminal SHOULD support the 'List Audio Files' command as specified in Section 4.4.8.1. |
| **MTT-2300** | The Terminal SHOULD support the 'Play File' command as specified in Section 4.4.8.2. |
| **MTT-2310** | The Terminal SHOULD support the 'Play Tone' command as specified in Section 4.4.8.3. |
| **MTT-2320** | The Terminal SHOULD support the 'Play DTMF' command as specified in Section 4.4.8.4. |
| **MTT-2330** | The Terminal SHOULD support the 'getPlayStatus' command as specified in Section 4.4.8.8. |
| **MTT-2340** | The Terminal SHOULD support the 'stop' command as specified in Section 4.4.8.5. |
| **MTT-2350** | The Terminal SHOULD support the 'Get Duration' command as specified in Section 4.4.8.6. |
| **MTT-2360** | The Terminal SHOULD support the 'Get Max Volume' command as specified in Section 4.4.8.7. |
| **MTT-2370** | The Terminal SHOULD support the 'Set Volume' command as specified in Section 4.4.8.8. |
| | Requirements refer to section 4.4.9 HTI PIM Service |
| **MTT-2380** | The Terminal MUST support the 'Import vCard' command as specified in Section 4.4.9.1. |
| **MTT-2390** | The Terminal MUST support the 'Import vCalendar' command as specified in Section 4.4.20.2. |
| **MTT-2400** | The Terminal MUST support the 'Delete Contact Entries' command without Entry ID as specified in Section 4.4.9.3 (Delete All Entries). |
| **MTT-2400.1** | The Terminal SHOULD support the 'Delete Contact Entries' command with Entry ID as specified in Section 4.4.9.3 (Delete specific Entry). |
| **MTT-2410** | The Terminal MUST support the 'Delete Calendar Entries' command as specified in Section 4.4.9.4. |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-2420** | The Terminal MAY support the ‘Add Notepad Memo’ command as specified in Section 4.4.9.5. |
| **MTT-2430** | The Terminal MAY support the ‘Add Notepad Memo from File’ command as specified in Section 4.4.9.6. |
| **MTT-2440** | The Terminal MAY support the ‘Delete All Notepad Memos’ command as specified in Section 4.4.9.7. |
| **MTT-2450** | The Terminal MUST support the ‘SIM Card Information’ command as specified in Section 4.4.9.8. |
| **MTT-2460** | The Terminal MUST support the ‘Import SIM Contacts’ command as specified in Section 4.4.9.9. |
| **MTT-2470** | The Terminal MUST support the ‘Delete SIM Contact’ command without Entry ID as specified in Section 4.4.9.10 (Delete All Contacts). |
| **MTT-2470.1** | The Terminal MUST support the ‘Delete SIM Contact’ command with Entry ID as specified in Section 4.4.9.10 (Delete specific Contact). |
| **MTT-2480** | If the Terminal has a browser, it MUST support the ‘Create Bookmark’ command as specified in Section 4.4.9.11. |
| **MTT-2490** | The Terminal MUST support the ‘Delete Bookmark’ command as specified in Section 4.4.9.12. |
| | Requirements refer to section 4.4.10 HTI Messaging Service |
| **MTT-2500** | The Terminal MUST support the ‘Add Sms’ command as specified in Section 4.4.10.1. |
| **MTT-2510** | The Terminal MUST support the ‘Add Mms’ command as specified in Section 4.4.10.2. |
| **MTT-2510.1** | The Terminal MAY support different types for MMS composing, e.g. with or without SMIL. |
| **MTT-2520** | The Terminal MUST support the ‘Add E-Mail’ command as specified in Section 4.4.10.3. |
| **MTT-2530** | The Terminal MAY support the ‘Add Ir Message’ command as specified in Section 4.4.10.4. |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-2540** | The Terminal MAY support the ‘Add BT Message’ command as specified in Section 4.4.10.5. |
| **MTT-2550** | The Terminal MAY support the ‘Add Smart Message’ command as specified in Section 4.4.10.6. |
| **MTT-2560** | The Terminal MAY support the ‘Add Audio Message’ command as specified in Section 4.4.10.7. |
| **MTT-2570** | The Terminal MUST support the ‘Delete Message’ command as specified in Section 4.4.10.8. |
| **MTT-2580** | The Terminal MUST support the ‘Delete Folder Content’ command as specified in Section 4.4.10.9. |
| **MTT-2590** | The Terminal MUST support the ‘Create Access Point’ command as specified in Section 4.4.10.10. |
| **MTT-2600** | The Terminal MUST support the ‘Delete Access Point’ command as specified in Section 4.4.10.11. |
| **MTT-2610** | The Terminal SHOULD support the ‘Create Destination’ command as specified in Section 4.4.10.12. |
| **MTT-2620** | The Terminal SHOULD support the ‘Delete Destination’ command as specified in Section 4.4.10.13. |
| **MTT-2630** | The Terminal SHOULD support the ‘Add to Destination’ command as specified in Section 4.4.10.14. |
| **MTT-2640** | The Terminal SHOULD support the ‘Remove from Destination’ command as specified in Section 4.4.10.15. |
| **MTT-2650** | The Terminal SHOULD support the ‘Set Default Connection’ command as specified in Section 4.4.10.16. |
| **MTT-2660** | The Terminal MUST support the ‘Create Mailbox’ command as specified in Section 4.4.10.17. |
| **MTT-2670** | The Terminal MUST support the ‘Delete Mailbox’ command as specified in Section 4.4.10.18. |
| **MTT-2680** | The Terminal MUST support the ‘Set Default SMS Center’ command as specified in Section 4.4.10.19. |

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-2690** | The Terminal MUST support the ‘Delete SMS Center’ command as specified in Section 4.4.10.20. |
| **MTT-2700** | The Terminal MUST support the ‘Set SMS Settings’ command as specified in Section 4.4.10.21. |
| **MTT-2710** | The Terminal MUST support the ‘Set MMS Settings’ command as specified in Section 4.4.10.22. |

## 3.5 General Connectivity Requirements

| Req. Id | Requirement |
|---|---|
| **MTT-2800** | If the Terminal is in a connected state through physical connection all standard Terminal applications and functionality SHALL be available via the Terminal user interface by users. This requirement is only applicable if phone mode has been selected for the data connection or HTI as per requirement MTT-0070 has been enabled. |
| **MTT-2810** | An option to deactivate battery recharging while connected SHALL be available (applicable e.g. for Terminals which automatically recharge when connected via USB). (see requirement MTT-1110 and chapter 8.1.2). |
| **MTT-2820** | It SHALL be possible to connect both audio and Test Computer cables simultaneously. |
| | AT-command syntax requirements |
| **MTT-2830** | For AT-commands, the Terminal MUST accept string parameters in quotation marks. |
| **MTT-2840** | For AT-commands, the Terminal MUST accept numeric parameters without quotation mark. |
| **MTT-2850** | For AT-commands, the Terminal MUST return string parameters in quotation marks. |
| **MTT-2860** | For AT-commands, the Terminal MUST return numeric parameters without quotation marks. |

## 3.6 SECURITY REQUIREMENTS

| REQ. ID | REQUIREMENT |
|---|---|
| **MTT-2900** | The test access to the AT command or HTI interface SHALL only be possible via a secured link. |
| **MTT-2900.1** | A cable MUST be supported to interconnect the Test Computer and the Terminal. |
| **MTT-2900.1.1** | The preferred connector to the Terminal SHALL be a Micro USB connector. |
| **MTT-2900.2** | The access to the AT command interface of the Terminal via SIM Application Toolkit (SAT, SIM-card) SHOULD be possible. |
| **MTT-2910** | Any access to the AT command or HTI interface over any other channels than specified in requirement MTT-2900 (e.g. GSM / UMTS or WLAN) MUST NOT be possible. |
| **MTT-2920** | The Terminal MUST comply with the OMTP ASF requirements [2]. |

# 4 HARMONISED TESTING INTERFACE – MESSAGING PROTOCOL

## 4.1 DOCUMENT CONTROL

### 4.1.1 DOCUMENTATION CONVENTIONS

Code is shown with `Courier New font.`

## 4.2 GENERAL

### 4.2.1 INTRODUCTION

HTI messaging is asynchronous and can be mapped over different communication media. There is no requirement for a reply to any incoming message unless it is an error situation. The message header specifies a target service by its plug-in unique identifier (UID) and the message body is passed directly to the service for processing. Each service uses it's own protocol and message format. The HTI framework provides some auxiliary services like sending a list of available service plugins, HTI framework version and status and support for security. Special reserved UID **`0x1020DEB6`** is used for these HTI messages.

This document describes protocol used for communication with HTI Framework and core plug-ins.

## 4.3 HTI PROTOCOL DESCRIPTION

### 4.3.1 MESSAGE STRUCTURE

A HTI message contains a fixed size header (14 bytes), optional header extension and variable length body. In a typical case, the message body is passed directly for processing to a service specified in the header.

**Table 1: HTI message structure**

| Field | Size(bytes) | Description |
|---|---|---|
| `ServicePluginUID` | 4 | Service plug-in UID in little-endian format, signed 32bit integer. |
| `MessageBodySize` | 4 | The size of the message body (in little-endian format, unsigned 32bit integer) |
| `Version` | 1 | Version of HTI message specification (1) |
| `Priority` | 1 | Message priority used when selecting messages for processing from the incoming queue. When the first bit (lowest) set it indicates the higheset priority used only for some system level requests. The second bit can be used by service plug-ins to have higher priority messages or organise the pair of data/control channels. Messages with the second bit set have higher priority over other messages to the |

| | | |
|---|---|---|
| | | same service plug-in. |
| `Flags` | 1 | The set of flags. Currently only the 1st bit is used:<br>1. `WrapFlag` indicates that message body is encoded by a security mechanism and needed to be decoded before passing to a target service (0=plain, 1=encoded) |
| `ExtensionSize` | 1 | The size of the extension field (Unsigned 8bit integer) |
| `CRC` | 2 | Checksum of previous fields (CCITT CRC 16 in little-endian format, unsigned 16 bit integer) |
| `ExtensionBody` | - | Extension fields reserved for new parameters |
| `MessageBody` | - | Message body, its content depends on addressed service. |

### *4.3.2 SECURITY SUPPORT*

The HTI protocol provides support for security mechanisms for communication. This security mechanism can be involved in following stages during communication:

- Communication establishment. At this stage, the HTI framework supports implementation of the authentication mechanism by providing the way to exchange tokens until security context is set with an authentication command (see 4.4.1.1).
- Data transfer. The HTI framework supports encrypting/decrypting for selected messages by specifying WrapFlag in the message header (see 4.3.1). Security manager will provide actual encoding implementation that may include integrity checking as well. **Currently, this feature is not supported and the flag is ignored.**

At the current stage, the HTI framework does not provide any security implementation, but it can be added later. Nevertheless, the AUTH command should be used to establish a security context and start service plug-ins use. The token parameter is currently ignored and can contain anything or can be omitted. The HTI framework will reply with the same message with an AUTH command to indicate that security context has been set. After this, the HTI framework will process all incoming messages. Until a security context has been established, an error message with the `ENotAuthorized` (see 4.3.3) error is sent back in reply to any incoming message. The following figure demonstrates the authentication procedure:

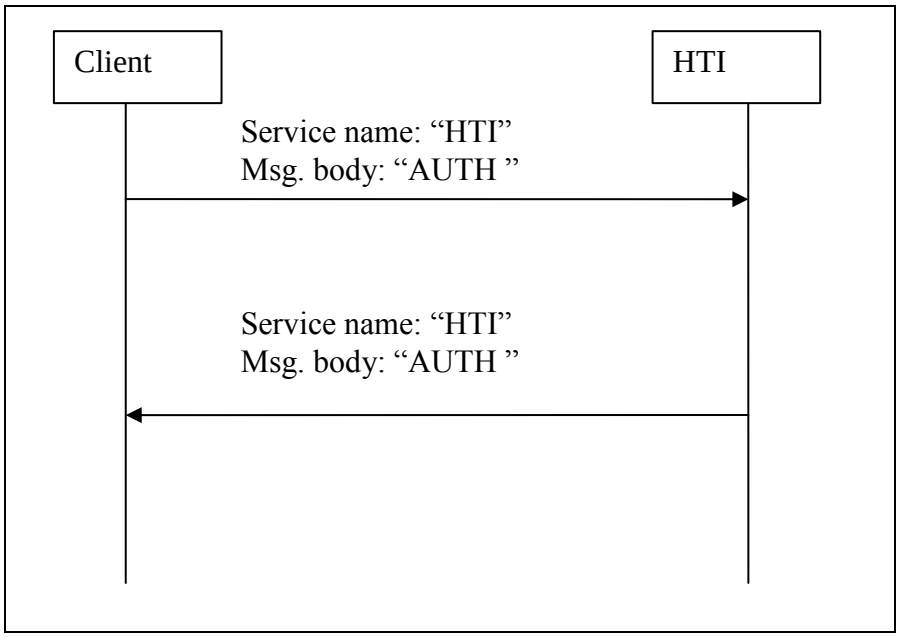



**Figure 2: Current authentication message exchange**

### *4.3.3 ERROR MESSAGES*

The HTI framework provides a unified approach to handling error messages. When an error occurs during request processing, an error message will be sent back. Target ServicePluginUID in the error message header is always set to HTI framework UID. The body contains the error description in the following format:

**Table 2: Error message structure**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xFF | 1 | Indicates error message (see 4.4.1) |
| HTIErrorCode | | 1 | Error code defined in Table 3 |
| ServiceErrorCode | | 4 | Either system-wide leave code or plug-in specific error code when plug-in handles error situation by itself (in little-endian format, signed 32bit integer) |
| ServicePluginUID | | 4 | ServicePluginUIDfrom the header of message processed (in little-endian format, signed 32bit integer) |
| ServiceErrorDescription | | variable | contains error description in case when error situation is handled by plug-in |

`ServiceErrorCode` and `ServiceErrorDescription` are specified only when an error occurs during message processing by a service plug-in. If a service plug-in just leaves than `ServiceErrorCode` will contain a leave code. In case a service plug-in reports an error to the framework by itself, the

`ServiceErrorCode` and `ServiceErrorDescription` content is defined by the plug-in.

Beside specified error messages, each plug-in can define its own error notification messages if necessary.

**Table 3: HTI error codes**

| ErrorName | Code | Description |
|---|---|---|
| EMessageTooBig | 1 | Message body size exceed predefined limit |
| EOutOfMemory | 2 | No sufficient amount of memory to process request |
| EServiceNotFound | 3 | Message is addressed to non-existent service |
| EServiceError | 4 | Error in plug-in while processing request |
| ENotAuthorized | 5 | Request is not authorized |
| EFailedUnwrap | 6 | Security manager failed to decode message (see 4.3.1) |

### 4.3.4 CHARACTER ENCODINGS

The next chapter lists the HTI services and their functions (commands). Some of the commands have character strings as parameters and different commands may use different encodings for the character strings. The expected encoding is mentioned in the parameter descriptions. The following sections briefly describe the used encodings.

#### 4.3.4.1 UCS-2

UCS-2 (2-byte Universal Character Set) is the character encoding used. In UCS-2 each character is represented with 16-bits so it is a fixed-length encoding. Thus UCS-2 can represent Unicode characters that are defined in Basic Multilingual Plane (BMP) that is character codes 0 – 65535 (U+0000 – U+FFFF). It should be noted that UCS-2 is not the same as UTF-16, since UTF-16 is a variable length encoding that can represent all of the Unicode characters (also those outside the BMP). However, the character codes in the BMP range are identical, so when working only with the character codes 0 – 65535 the encodings UCS-2 and UTF-16 are interchangeable.

An example (using Java) on how to encode a character string to a byte array using UCS-2/UTF-16 little-endian encoding:

```
byte[] data = "t€st string".getBytes( "UTF-16LE" );
```

#### 4.3.4.2 UTF-8

UTF-8 (8-bit Unicode Transformation Format) encodes each character in one to four bytes, so it is a variable length encoding. UTF-8 is able to represent any character defined in the Unicode standard. The ASCII characters are

encoded as one byte in UTF-8 and the characters codes are also the same, so UTF-8 and ASCII encodings of a character string that contains only ASCII characters are identical. Characters in Unicode range U+0080 – U+07FF take two bytes and the rest of the characters in BMP take three bytes.

An example (using Java) on how to encode a character string to a byte array using UTF-8 encoding:

```
byte[] data = "t€st string".getBytes( "UTF-8" );
```

#### *4.3.4.3 ISO 8859-1*

When in this document “8-bit text” is mentioned it refers to ISO 8859-1 encoding where each character is encoded with one 8-bit byte. IS0 8859-1 contains 191 characters of Latin alphabet. The rest of the character codes are reserved for non-printable control characters. The encoding is backwards compatible with ASCII since the character codes of ASCII maps to the same characters also in ISO 8859-1. The ISO 8859-1 just adds more characters in the byte range A0 – FF.

An example (using Java) on how to encode a character string to a byte array using ISO 8859-1 encoding:

```
byte[] data = "tëst string".getBytes( "ISO-8859-1" );
```

Note that ISO 8859-1 does not contain the € character or any other character that Windows-1252 encoding defines in the byte range 80 – 9F. That range in ISO 8859-1 contains non-printable control characters.

## 4.4 HTI SERVICES

HTI provides several core test services described in the table below.

**Table 4: HTI services**

| Service Name | UID | Description |
|---|---|---|
| HTI | 0x1020DEB6 | HTI framework system services:<br>• send available service list<br>• send HTI framework version<br>• security support<br>• device reboot |
| Echo | 0x1020DEBF | Service to test HTI itself, that the HTI service is up and running and communication works properly. |
| Key pressing | 0x1020DEC1 | Key press emulation service provides following functions:<br>• single key press<br>• single key hold / release<br>• key sequence press |
| Screen capture | 0x1020DEC3 | Screen capture makes screenshot on request and sends it back:<br>• capture full screen<br>• capture screen region |
| Applications control | 0x1020DEC7 | The service allows for example:<br>• start application by its name or UID<br>• start application with document<br>• check application status (running or not)<br>• stop application<br>• start / stop processes<br>• get process exit code |
| FTP | 0x1020DEC5 | The service allow to copy/retrieve files to/from a device:<br>• copy file (STOR)<br>• retrieve file (RETR)<br>• list files (LIST)<br>• create directory (MKD)<br>• delete directory (RMD)<br>• delete file (DELE) |
| SysInfo | 0x10210CC7 | The service allow to retrieve different device information:<br>• HAL attributes (see HAL class)<br>• IMEI code<br>• SWVersion (see SysUtil class)<br>• LangVersion (see SysUtil class)<br>• SWLangVersion (see SysUtil class)<br>• UserAgentString (see SysUtil class) |
| Audio Control | 0x10210CCB | Service for playing audio tones and samples:<br>• List audio sample files<br>• Play tones with specified frequency<br>• Play DTMF tones<br>• Play audio sample files of different formats (wav, mid, mp3, |

| | | |
|---|---|---|
| | | awb…)<br>• Control volume |
| PIM service | 0x10210CCD | Service for adding and removing contact and calendar entries. |
| Messages service | 0x10210CCF | Service for adding and removing messages. |

The following subchapters provide protocol and message format description for each service.

Whenever possible, all service use similar structure for requests. The first byte contains the command code with the command parameters following.

### 4.4.1 HTI SERVICE

The HTI service provides some HTI framework-level auxiliary functions that can not be addressed to or implemented as a separate service plug-in. The following commands are supported:

**Table 5: HTI commands**

| CommandName | CommandCode |
|---|---|
| HtiAuthentication | 0x01 |
| HtiVersion | 0x02 |
| HtiServiceList | 0x03 |
| HtiStop | 0x04 |
| HtiReboot | 0x05 |
| HtiFormat | 0x06 |
| HtiReset | 0x07 |
| HtiShowConsole | 0x08 |
| HtiHideConsole | 0x09 |
| HtiInstanceId | 0x0A |
| HtiDebugPrint | 0x0B |
| HtiError | 0xFF |

#### 4.4.1.1 Authentication

The AUTH command is used for bidirectional exchange of tokens during authentication process. The token content is opaque for the HTI framework and only passed to the security implementation that makes the decision when the authentication process is over and the result. The first byte indicates the authentication command and the rest of HTI message treated as a token with a variable length.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x01 | 1 | HtiAuthentication |

| | | | |
|---|---|---|---|
| Token | | variable | Token value |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x01 | 1 | `HtiAuthentication` |
| Token | | variable | Token value |

### *4.4.1.2 HTI version*

Returns the HTI framework version.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x02 | 1 | `HtiVersion` |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| MajorVersion | | 1 | Major version |
| MinorVersion | | 1 | Minor version |

### *4.4.1.3 Service list*

Returns the list of installed service plug-ins. A list item record is 128 bytes long and contains plug-in UID and display name. The reply message contains the sequential record list.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x03 | 1 | `HtiServiceList` |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| ServicePluginUID | | 4 | Service plug-in UID |
| ServicePluginName | | 124 | Service display name 8-bit encoded |

#### 4.4.1.4 Stop

Stops HTI framework and shuts it down.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x04 | 1 | `HtiStop` |

#### 4.4.1.5 Reboot

Reboots the device.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x05 | 1 | `HtiReboot` |

#### 4.4.1.6 Restore factory settings

Performs a factory settings reset (either normal or deep).

It will clear the entire memory of the device and reinitialise it with the default folder structure. The deep mode is allowed only if HTI is running from ROM.

This command will cause the device to reboot.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x06 | 1 | `HtiFormat` |
| Mode | 0x00 or 0x01 | 1 | `The restore factory settings mode (0 = normal, 1 = deep)` |

#### 4.4.1.7 Reset

Reset the HTI framework by unloading all service plug-ins and clearing message queues.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x07 | 1 | `HtiReset` |

#### *4.4.1.8 Show Console*

Opens a console where HTI and its test services may print information to.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x08 | 1 | `HtiShowConsole` |

**Response**

Byte 0x00 on success or error message (see 4.3.3).

#### *4.4.1.9 Hide console*

Hides (deletes) the open HTI console.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x09 | 1 | `HtiHideConsole` |

**Response**

Byte 0x00 on success or error message (see 4.3.3).

#### *4.4.1.10 Instance ID*

Returns the unique instance ID of a current HTI instance.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0A | 1 | `HtiInstanceId` |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| InstanceID | | 4 | The instance ID value as 32-bit unsigned little-endian integer. |

#### 4.4.1.11 Debug print

Writes a message to the device's debug port (RDebug).

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0B | 1 | `HtiDebugPrint` |
| Message | | variable | The text to be printed to debug port. Encoded as 8-bit text. |

### Response

Byte 0x00 on success or error message (see 4.3.3).

### 4.4.2 ECHO SERVICE

Echo service is intended for testing HTI communication media. It immediately sends back anything it receives inside message body.

### 4.4.3 KEY EVENTSSERVICE

This service allows emulation of key pressing for UI application via Window Server.

Scancode in the following specification defines a keyboard key. It is 16-bit unsigned integer. Each device or device family maps these to their native keyboard keys. For example, button "1" on the phone is defined typically as EStdKeyNkp1 = 0x89.

Key pressing service returns an ok message after the command has been executed successfully. Format of ok message:

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xFF | 1 | `Notification of successful command execution.` |

On an error the HTI framework's error method is used (see 4.3.3). General error codes and descriptions for key pressing service are defined below.

**General error codes**

| Error code | Error description |
|---|---|
| < 0 | Indicates internal error in the service. |
| 0x01 | Missing key event command – the message was empty |
| 0x02 | Invalid key event service command – command not recognised |
| 0x03 | Key event service is busy. It is possibly executing a long running typetext function or long key press function. |
| 0x04 – 0x7F | *Reserved for future* |

Should the need arise to send more key events, ensure there is an adequate delay between pressing each key (or holding key down, or releasing key up). The delay needed depends on the system, but typically 50 milliseconds should be enough.

Type text functionality takes care of the appropriate interval.

#### *4.4.3.1 Single key press*

This emulates pressing a single key on the keyboard. At a low level this simulates pressing the keyboard key down and immediately releases it.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x01 | 1 | Command for single key press |
| Scancode | | 2 | 16-bit unsigned integer specifying keyboard key to press. |

**Errors**

| Error code | Error description |
|---|---|
| 0x80 | Invalid scancode. The length of the scancode field was less or more than 2 bytes. |

#### *4.4.3.2 Key down*

Emulates pressing a keyboard key down without releasing it. Using this function should be paired with Key up function.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x02 | 1 | Command for pressing key down. |
| Scancode | | 2 | 16-bit unsigned integer specifying keyboard key to press. |

**Errors**

| Error code | Error description |
|---|---|
| 0x80 | Invalid scancode. The length of the scancode field was less or more than 2 bytes. |

#### 4.4.3.3 Key up

Release a key that was held down by the Key down function.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x03 | 1 | Command for releasing key. |
| Scancode | | 2 | 16-bit unsigned integer specifying keyboard key to release. |

**Errors**

| Error code | Error description |
|---|---|
| 0x80 | Invalid scancode. The length of the scancode field was less or more than 2 bytes. |

#### 4.4.3.4 Type text

This command emulates typing unicode characters to the topmost window.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x04 | 1 | Command for type text function. |
| Array of Unicode characters | | varies | Array of 16-bit unsigned integers specifying Unicode characters (UCS-2) to type. |

**Errors**

| Error code | Error description |
|---|---|
| 0x90 | Array of Unicode characters is empty |
| 0x91 | Invalid array of Unicode characters. Array should be divisible by two, because each character is UCS-2. |

#### 4.4.3.5 Long key press

This command emulates pressing and holding down a single key and releasing it after a specified time. The functionality is the same as using the “Key down” and “Key up” commands consecutively but this command automates both events at a specified interval.

Note that when a long key press is active the key event service is busy and it won’t accept any other commands before the key press ends (the “ok” response message is sent when the key is released).

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x05 | 1 | Command for long key press |
| Scancode | | 2 | 16-bit unsigned integer specifying |

| | | | |
|---|---|---|---|
| | | | keyboard key to press. |
| Time | | 2 | 16-bit unsigned integer specifying the time to hold the key down. Time is specified in milliseconds. |

**Errors**

| Error code | Error description |
|---|---|
| 0x80 | Invalid scancode. The length of the command message was not 5 bytes as stated in the Syntax table above. |

### *4.4.3.6 Type text password*

The command emulates typing unicode characters to topmost password window. A special command for this is needed because normal 'type text' -command will result in all characters appearing twice.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x06 | 1 | Command for type text password function. |
| Array of Unicode characters | | Varies | Array of 16-bit unsigned integers specifying Unicode characters (UCS-2) to type. |

**Errors**

| Error code | Error description |
|---|---|
| 0x90 | Array of Unicode characters is empty |
| 0x91 | Invalid array of Unicode characters. Array should be divisible by two, because each character is UCS-2. |

### *4.4.3.7 Key press sequence*

This command emulates pressing a sequence of keys with a given interval between them and time that each key is held down. The time that each key is held down can be zero, when they are released immediately after pressing them down. The minimum interval between key presses is 50 ms.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x07 | 1 | Command for key press sequence function. |
| Time | | 2 | 16-bit unsigned integer specifying the time to hold each key down. Time is specified in milliseconds. |
| Interval | | 2 | 16-bit unsigned integer specifying the time between each key. Time is specified in milliseconds. |

| | | | |
|---|---|---|---|
| Scancode | | 2 | 16-bit unsigned integer specifying keyboard key to press. |
| (The scancodes for the rest of the keys in the sequence) | | | |

**Errors**

| Error code | Error description |
|---|---|
| 0x90 | Array of Unicode characters is empty |
| 0x91 | Invalid array of Unicode characters. Array should be divisible by two, because each character is UCS-2. |

### *4.4.3.8 Tap screen*

This command emulates tapping one point in the touch screen. Taps can be repeated and the interval specified.

All parameters after the command code are specified as 16-bit little-endian unsigned integers.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x10 | 1 | Command for tap screen. |
| X coordinate | | 2 | The screen X coordinate to tap. |
| Y coordinate | | 2 | The screen Y coordinate to tap. |
| Time to hold down | | 2 | The time in milliseconds how long the pointer is kept down before lifting up. |
| Tap count | | 2 | How many times to repeat the tap to the same point. |
| Pause between taps | | 2 | The time in milliseconds how long pause to keep between consecutive taps (if more than one). |

**Errors**

| Error code | Error description |
|---|---|
| 0x9A | The parameters given to the command are invalid. |

### *4.4.3.9 Tap and drag*

This command emulates a drag and drop operation using a straight line between two given points. The length of time that the drag takes can be specified.

All parameters after the command code are specified as 16-bit little-endian unsigned integers.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x11 | 1 | Command for tap and drag. |
| Start X coordinate | | 2 | The screen X coordinate where the pointer is pressed down. |
| Start Y coordinate | | 2 | The screen Y coordinate where the pointer is pressed down. |
| End X coordinate | | 2 | The screen X coordinate where the pointer is lifted up. |
| End Y coordinate | | 2 | The screen Y coordinate where the pointer is lifted up. |
| Drag time | | 2 | The time in milliseconds that the drag takes. |

**Errors**

| Error code | Error description |
|---|---|
| 0x9A | The parameters given to the command are invalid. |

### *4.4.3.10 Tap and drag multipoint*

This command emulates a drag operation via multiple points thus enabling drawing of curvy lines. One command can contain one or many lines.

All parameters after the command code are specified as 16-bit little-endian unsigned integers.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x12 | 1 | Command for tap and drag multipoint. |
| Time between points | | 2 | The time in milliseconds it takes to drag from one point to the next, so this controls how fast the dragging happens. |
| Time between lines | | 2 | The time in milliseconds that a pause is kept between ending the drawing of one line and starting another. |
| Point count of first line | | 2 | How many points (x and y coordinate pairs) the first line contains. |
| First X coordinate | | 2 | The screen X coordinate from where the line begins. |
| First Y coordinate | | 2 | The screen Y coordinate from where the line begins. |
| Second X coordinate | | 2 | The screen X coordinate of the second point where to drag the pointer. |
| Second Y coordinate | | 2 | The screen Y coordinate of the second point where to drag the pointer. |
| | | | The X and Y coordinates repeated as many times as specified in the point count. The pointer is lifted up in the last |

| | | | |
|---|---|---|---|
| | | | coordinate position. |
| Point count of the second line | | 2 | How many points (x and y coordinates) the following line contains (if drawing more than one lines with the same command). |
| | | | The X and Y coordinates repeated as many times as specified in the point count. Then the point count of the next line, point coordinates, etc. |

**Errors**

| Error code | Error description |
|---|---|
| 0x9A | The parameters given to the command are invalid. |

### *4.4.3.11 Pointer down*

This command emulates pressing a pointer down in specified point without releasing it. Using this command should be paired with Pointer up command.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x13 | 1 | Command for pressing pointer down. |
| X coordinate | | 2 | The screen X coordinate where the pointer is pressed down. |
| Y coordinate | | 2 | The screen Y coordinate where the pointer is pressed down. |

**Errors**

| Error code | Error description |
|---|---|
| 0x9A | The parameters given to the command are invalid. |

### *4.4.3.12 Pointer up*

This command emulates lifting the pointer up (that was pressed down with the Pointer down command). The coordinates of the point where the pointer is lifted up can be same or different than what was given to the Pointer down command. In case of different coordinates, the pointer drags to the pointer up coordinates before lift.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x14 | 1 | Command for lifting the pointer up. |
| X coordinate | | 2 | The screen X coordinate where the pointer is lifted up. |
| Y coordinate | | 2 | The screen Y coordinate where the pointer is lifted up. |

**Errors**

| Error code | Error description |
|---|---|
| 0x9A | The parameters given to the command are invalid. |

### 4.4.4 SCREEN CAPTURING SERVICE

This service allows capturing either full screen or its region and sends back as image file. By default, bitmap format is used, but another format can be specified with a supported MIME type parameter. The service contains also text recognition related functionalities.

The following commands are supported:

**Table 6: Screen capture commands**

| CommandName | CommandCode |
|---|---|
| Screen | 0x01 |
| ScreenRegion | 0x02 |
| ScreenZip | 0x03 |
| ScreenRegionZip | 0x04 |
| TextRecognition | 0x10 |
| TextRecognition_u | 0x11 |
| TextBitmap | 0x12 |
| TextBitmap_u | 0x13 |
| ScreenSeries | 0x21 |
| ScreenRegionSeries | 0x22 |
| ScreenZipSeries | 0x23 |
| ScreenRegionZipSeries | 0x24 |
| SelectScreen | 0x30 |
| ScreenMode | 0x3A |
| ScreenDelta | 0x81 |
| ScreenDeltaRegion | 0x82 |
| ScreenDeltaZip | 0x83 |
| ScreenDeltaRegionZip | 0x84 |
| DeltaReset | 0x85 |

Optionally, colour depth of output screenshot can be specified. Values from TDisplayMode enumeration (gdi.h) are used. If ENone (0) is specified, then current display mode is used. Following values are currently supported (from gdi.h):

**Table 7: Display modes**

| CommandName | CommandCode |
|---|---|
| ENone | 0x00 |
| EGray2 | 0x01 |
| EGray4 | 0x02 |
| EGray16 | 0x03 |
| EGray256 | 0x04 |
| EColor16 | 0x05 |
| EColor256 | 0x06 |
| EColor64K | 0x07 |
| EColor16M | 0x08 |
| ERgb | 0x09 |
| EColor4K | 0x0A |
| EColor16MU | 0x0B |
| EColor16MA | 0x0C |

Text recognition commands can accept font attributes. The following table contains attribute constants that could be combined to change a default attribute set. Default attributes are normal stroke weight, normal posture, normal print position, anti-aliased bitmap glyph type. Note, that superscript and subscript should not be combined.

**Table 8: Font attributes**

| AttributeName | AttributeCode |
|---|---|
| Bold stroke weight | 0x01 |
| Italic posture | 0x02 |
| Not anti-aliased bitmap glyph type | 0x04 |
| Superscript | 0x08 |
| Subscript | 0x18 |

Text recognition command uses following response codes:

**Table 9: Text recognition command response codes**

| ResponseName | ResponseCode |
|---|---|
| Ok | 0xF0 |
| NotFound | 0xF1 |

The screen mode command uses the following screen orientation values from CFbsBiGc::TGraphicsOrientation enumeration:

**Table 10: Screen rotation values**

| RotationName | RotationCode |
|---|---|
| EGraphicsOrientationNormal | 0x00 |
| EGraphicsOrientationRotated90 | 0x01 |
| EGraphicsOrientationRotated180 | 0x02 |
| EGraphicsOrientationRotated270 | 0x03 |

#### *4.4.4.1 Full screen capture*

Returns a screenshot of the whole screen. By default, the image is sent in uncompressed bitmap format. Another format can be specified in the `ImageMIMEType` parameter by its MIME type (e.g. “image/png”, “image/gif”, “image/jpeg”). If a message contains anything after the first byte, it is treated as a MIME type, otherwise if a message contains only one byte, the default format is used.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x01<br>0x03 | 1 | `Screen, ScreenZip` |
| ColorDepth | | 1 | Value from TDisplayMode, optional |
| ImageMIMEType | | variable | Registered image MIME type as 8-bit encoded text, optional |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Image | | variable | Data of the returned image. |

#### *4.4.4.2 Screen region capture*

Returns screenshot of a screen region. By default, image is sent in uncompressed bitmap format. Another format can be specified in the `ImageMIMEType` parameter by its MIME type (e.g. “image/png”, “image/gif”, “image/jpeg”). Notice, that bottom right coordinate of the specified region is exclusive. E.g., to specify a 32 pixels by 32 pixels region in the top left corner of the screen, (0,0)x(32,32) coordinates are used.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x02<br>0x04 | 1 | `ScreenRegion, ScreenRegionZip` |
| Top left x coordinate | | 2 | Top left corner of the region (inclusive) as little-endian unsigned 16-bit integer |
| Top left y coordinate | | 2 | … |
| Bottom right x coordinate | | 2 | Bottom right corner of the region (exclusive) |

| | | | |
|---|---|---|---|
| | | | as little-endian unsigned 16-bit integer |
| Bottom right y coordinate | | 2 | … |
| ColorDepth | | 1 | Value from TDisplayMode, optional |
| ImageMIMEType | | variable | Registered image MIME type as 8-bit encoded text |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Image | | variable | Data of the returned image. |

### *4.4.4.3 Text Recognition*

The text recognition command defines that the text in question is currently displayed on screen using one of provided fonts.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x10<br>0x11 | 1 | TextRecognition, TextRecognition _u |
| TextLength | | 1 | Text length in characters |
| Text | | variable | Text either encoded as 8-bit text or as UCS-2 for Unicode variant command |
| NumberOfFonts | | 1 | Number of font specifications, should be at least 1. |
| TypefaceNameLength | | 1 | Typeface name length. |
| TypefaceName | | variable | Typeface name either encoded as 8-bit text or as UCS-2 for Unicode variant command |
| FontHeight | | 2 | Font height in pixels as unsigned little-endian 16bit integer |
| FontStyle | | 1 | Font style as combination of font attributes (see Table 8) |

Fields TypefaceNameLength, TypefaceName, FontHeight and FontStyle are repeated NumberOfFonts time.

**Return**

Ok, NotFound or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0<br>0xF1 | 1 | Ok, NotFound |
| Top left X | | 2 | The x-coordinate of the top left corner (inclusive) of the found text. Encoded as unsigned little-endian 16bit integer. For Ok command |

| | | | |
|---|---|---|---|
| Top left Y | | 2 | Text y-coordinate of the top left corner (inclusive) of the found text. Encoded as unsigned little-endian 16bit integer. For Ok command |
| Bottom right X | | 2 | The x-coordinate of the bottom right corner (exclusive) of the found text. Encoded as unsigned little-endian 16bit integer. For Ok command |
| Bottom right Y | | 2 | The y-coordinate of the bottom right corner (exclusive) of the found text. Encoded as unsigned little-endian 16bit integer. For Ok command |
| FontIndex | | 1 | Font of the found text as index to the font specifications passed in the request. For Ok command. |

#### *4.4.4.4 Text bitmap*

This command generates a bitmap with text in question using the specified font. By default, image is sent in uncompressed bitmap format. Another format can be specified in the `ImageMIMEType` parameter by its MIME type (e.g. “image/png”, “image/gif”, “image/jpeg”).

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x12<br>0x13 | 1 | `TextBitmap, TextBitmap_u` |
| ColorDepth | | 1 | Value from TDisplayMode, optional |
| ImageMIMEType length | | 1 | ImageMIMEType length |
| ImageMIMEType | | variable | Registred image MIME type as 8-bit encoded text, optional |
| TextLength | | 1 | Text length in characters |
| Text | | variable | Text either encoded as 8-bit text or as UCS-2 for Unicode variant command |
| TypefaceNameLength | | 1 | Typeface name length. |
| TypefaceName | | variable | Typeface name either encoded as 8-bit text or as UCS-2 for Unicode variant command |
| FontHeight | | 2 | Font height in pixels as unsigned little-endian 16bit integer |
| FontStyle | | 1 | Font style as combination of font attributes (see Table 8) |
| Font color | | 4 | Color in the 0x00BBGGRR form as unsigned little-endian 32bit integer |
| Background color | | 4 | Color in the 0x00BBGGRR form as unsigned little-endian 32bit integer |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Image | | variable | Data of the returned image. |

### *4.4.4.5 Full screen capture in series*

The command captures series of screenshots of the whole screen. By default, image is sent in uncompressed bitmap format. Another format can be specified in the `ImageMIMEType` parameter by its MIME type (e.g. “image/png”, “image/gif”, “image/jpeg”). If a message contains anything after the `ColorDepth` byte, it is treated as a MIME type, otherwise if a message contains only one byte, the default format is used. List of full paths to screenshot images is returned in the response.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x21<br>0x23 | 1 | `ScreenSeries, ScreenZipSeries` |
| Duration | | 4 | Time to take sreenshots in microseconds |
| Interval | | 4 | Time in between shots in mircoseconds. Encoding time is not included in this. |
| ColorDepth | | 1 | Value from TDisplayMode, optional |
| ImageMIMEType | | variable | Registered image MIME type as 8-bit encoded text, optional |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| PathNameLenght | | 1 | Length of PathName in bytes |
| PathName | | Variable | Full path to screenshot file as 8-bit text |

### *4.4.4.6 Screen region capture in series*

The command captures series of screenshots of the specified region of the screen. By default, image is sent in uncompressed format. Another format can be specified in the `ImageMIMEType` parameter by its MIME type (e.g. “image/png”, “image/gif”, “image/jpeg”). If a message contains anything after the `Bottom right y coordinate` word, it is treated as a MIME type, otherwise if a message contains only one byte, the default format is used. List of full paths to screenshot images is returned in the response.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x22<br>0x24 | 1 | `ScreenRegionSeries, ScreenRegionZipSeries` |

| | | | |
|---|---|---|---|
| Duration | | 4 | Time to take sreenshots in microseconds |
| Interval | | 4 | Time in between shots. Encoding time is not included in this. |
| ColorDepth | | 1 | Value from TDisplayMode, optional |
| Top left x coordinate | | 2 | Top left corner of the region (inclusive) as little endian unsigned 16-bit integer |
| Top left y coordinate | | 2 | … |
| Bottom right x coordinate | | 2 | Bottom right corner of the region (exclusive) as little endian unsigned 16-bit integer |
| Bottom right y coordinate | | 2 | … |
| ImageMIMEType | | variable | Registered image MIME type as 8-bit encoded text, optional |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| PathNameLength | | 1 | Length of PathName in bytes |
| PathName | | Variable | Full path to screenshot file as 8-bit text |

### *4.4.4.7 Select screen*

This command can be used to change the screen where the screenshot commands are targeted. This command is only applicable if the device has more than one screen.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x30 | 1 | `SelectScreen` |
| Screen number | | 1 | The index number of the screen to use. Usually the main display is 0 and cover UI display is 1. |

**Return**

Byte 0x30 (the command code) if OK, otherwise an error message.

### *4.4.4.8 Screen mode*

This command can be used to query the attributes of the currently selected screen. If there is more than one screen in the device, the attributes of other screens can be queried by first changing the currently selected screen with the Select Screen command.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x3A | 1 | ScreenMode |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| This screen number | | 1 | The index number of the screen whose attributes are returned. |
| Screen width | | 2 | The width of the screen in pixels as little endian unsigned 16-bit integer. |
| Screen height | | 2 | The height of the screen in pixels as little endian unsigned 16-bit integer. |
| Screen rotation | | 1 | The rotation of the screen. 0 = normal, 1 = 90°, 2 = 180°, 3 = 270° |
| Display mode | | 1 | The display mode of the screen. Value from TDisplayMode (see Table 7). |
| Focus screen number | | 1 | The index number of the screen that currently has focus.<br>Note that this can be different than the currently selected screen. |

#### *4.4.4.9 Full screen delta capture*

This command captures only the changed part of the screen. The current screen is compared to the screen saved by previous 'delta' command. Rectangle coordinates of the changed region and corresponding part of the screen is returned in the response.

The whole screen and coordinates for maximum sized rectangle are returned if the previously saved image:

- does not exist
- its display mode does not match the current screen
- its size is different than the current screen

If the previous image is exactly the same as current image then the rectangle coordinates are set to zero and no image is returned.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x81<br>0x83 | 1 | ScreenDelta, ScreenDeltaZip |
| ColorDepth | | 1 | Value from TDisplayMode, optional |
| ImageMIMEType | | variable | Registered image MIME type as 8-bit |

| | | | |
|---|---|---|---|
| | | | encoded text, optional |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Top left x coordinate | | 2 | Top left corner of the region (inclusive) as little-endian unsigned 16-bit integer |
| Top left y coordinate | | 2 | … |
| Bottom right x coordinate | | 2 | Bottom right corner of the region (exclusive) as little-endian unsigned 16-bit integer |
| Bottom right y coordinate | | 2 | … |
| Image | | variable | Data of the returned image. Optional |

#### *4.4.4.10 Screen region delta capture*

Functions like the 'Full screen delta capture' command but all returned coordinates are relative to the given region.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x82<br>0x84 | 1 | `ScreenDeltaRegion`,<br>`ScreenDeltaRegionZip` |
| Top left x coordinate | | 2 | Top left corner of the region (inclusive) as little-endian unsigned 16-bit integer |
| Top left y coordinate | | 2 | … |
| Bottom right x coordinate | | 2 | Bottom right corner of the region (exclusive) as little-endian unsigned 16-bit integer |
| Bottom right y coordinate | | 2 | … |
| ColorDepth | | 1 | Value from TDisplayMode, optional |
| ImageMIMEType | | variable | Registered image MIME type as 8-bit encoded text |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Top left x coordinate | | 2 | Top left corner of the region (inclusive) as little-endian unsigned 16-bit integer |
| Top left y coordinate | | 2 | … |
| Bottom right x coordinate | | 2 | Bottom right corner of the region (exclusive) as little-endian unsigned 16-bit integer |
| Bottom right y coordinate | | 2 | … |
| Image | | variable | Data of the returned image. Optional |

#### 4.4.4.11 Screen delta reset

This command resets the delta screen capture so that the next call to delta capture returns the full screen (or region if using region delta capture) as if it would be the first call to delta capture.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x85 | 1 | DeltaReset |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x85 | 1 | Delta reset command processed ok. |

### 4.4.5 APPLICATION CONTROL SERVICE

This service allows start and stop applications and programs (exe) identified either by name or UID. The plug-in support the following set of commands:

**Table 11: Application control commands**

| CommandName | CommandCode |
|---|---|
| StartProcess | 0x02 |
| StartProcess_u | 0x03 |
| StatusProcess | 0x04 |
| StatusProcess_u | 0x05 |
| StatusProcessId | 0x06 |
| StopProcess | 0x08 |
| StopProcess_u | 0x09 |
| StopProcessId | 0x0A |
| ListProcesses | 0x0C |
| ListProcesses_u | 0x0D |
| StartProcessRetVal | 0x0E |
| StartProcessRetVal_u | 0x07 |
| GetProcessExitCode | 0x0B |
| StartApp | 0x10 |
| StartApp_u | 0x11 |
| StartAppUid | 0x12 |
| StartAppUid_u | 0x13 |
| StartDoc | 0x14 |
| StartDoc_u | 0x15 |
| StatusApp | 0x16 |

| | |
|---|---|
| StatusApp_u | 0x17 |
| StatusDoc | 0x18 |
| StatusDoc_u | 0x19 |
| StatusAppUid | 0x1A |
| StopApp | 0x1C |
| StopApp_u | 0x1D |
| StopDoc | 0x1E |
| StopDoc_u | 0x1F |
| StopAppUid | 0x20 |
| ListApps | 0x24 |
| ListApps_u | 0x25 |
| Install | 0x30 |
| Install_u | 0x31 |
| UnInstall | 0x32 |

Note that the first bit is used to indicate unicode variant.

**Table 12: Application control response codes**

| ResponseName | ResponseCode |
|---|---|
| Ok | 0xF0 |
| NotFound | 0xF1 |
| AlreadyRunning | 0xF2 |
| AlreadyStopped | 0xF3 |
| Running | 0xF4 |
| Killed | 0xF5 |
| Panic | 0xF6 |

String parameters for all commands specified as string length byte with string content bytes following. String length is specified in characters.

#### *4.4.5.1 Start process*

Start process defined by exe file name. If an exe file located in standard system folder, e.g. c:\system\programs, than only file name can be specified. If an exe file located in some separate folder, full path should be used with the file name.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x02<br>0x03 | 1 | StartProcess， StartProcess_u |
| ProgramNameLength | | 1 | Program name length in characters |
| ProgramName | | variable | Program full name either encoded as 8-bit text or as UCS-2 for Unicode variant command |
| CommandLineArgumentLength | | 1 | Command line argument string length in characters |
| CommandLineArgument | | variable | Command line argument string either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

Ok, NotFound or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0<br>0xF1 | 1 | Ok, NotFound |
| ProcessId | | 4 | Process id as unsigned little-endian 32bit integer for Ok command |

#### *4.4.5.2 Stop process*

Stop process defined by a match pattern that can identify a process.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x08<br>0x09 | 1 | StopProcess， StopProcess _u |
| ProcessNameLength | | 1 | Process name length in characters |
| ProcessName | | variable | Math pattern either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

Ok, NotFound or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0<br>0xF1 | 1 | Ok, NotFound |

### 4.4.5.3 Stop process by id

Stop process using the process id returned during process start.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0A | 1 | StopProcessId |
| ProcessId | | 4 | Process Id as unsigned little-endian 32bit integer |

**Return**

Ok, NotFound or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0<br>0xF1 | 1 | Ok, NotFound |

### 4.4.5.4 Get process status

Check the status of a process defined by a match pattern.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x04<br>0x05 | 1 | StatusProcess, StatusProcess _u |
| ProcessNameLength | | 1 | Process name length in characters |
| ProcessName | | variable | Math pattern either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

NotFound, Running or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF1<br>0xF4 | 1 | NotFound or Running |

### 4.4.5.5 Get process status by id

Check the status of a process using the process id returned during process start.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x06 | 1 | StatusProcessId |
| ProcessId | | 4 | Process Id as unsigned little-endian 32bit integer |

**Return**

NotFound, Running or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF1<br>0xF4 | 1 | NotFound or Running |

### *4.4.5.6 Start application by name*

Start an application defined by its name. Optional `DocumentName` parameter can be specified to pass it to the application started. The `ApplicationName` parameter can be a full path to the application file (including drive letter) or just the application filename (with or without the file extension).

If the application that is being started is already running, it will be brought to the foreground.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x10<br>0x11 | 1 | StartApp, StartApp_u |
| ApplicationNameLength | | 1 | Application name length in characters |
| ApplicationName | | variable | Application name either encoded as 8-bit text or as UCS-2 for Unicode variant command |
| DocumentNameLength | | 1 | Document name length in characters |
| DocumentName | | variable | Document full name (including drive letter and path) either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

Ok, NotFound, AlreadyRunning or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0<br>0xF1<br>0xF2 | 1 | Ok, NotFound, AlreadyRunning |
| ThreadId | | 4 | Application's main thread id as unsigned little-endian 32bit integer for Ok command |

### *4.4.5.7 Start application by UID*

Start an application defined by its UID. Optional `DocumentName` parameter can be specified to pass it to the application started.

If the application that is being started is already running, it will be brought to the foreground.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x12<br>0x13 | 1 | StartAppUid, StartAppUid_u |
| ApplicationUid | | 4 | Application UID as signed little-endian 32bit integer |
| DocumentNameLength | | 1 | Document name length in characters |
| DocumentName | | variable | Document full name (including drive letter and path) either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

Ok, NotFound, AlreadyRunning or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0<br>0xF1<br>0xF2 | 1 | Ok, NotFound, AlreadyRunning |
| ThreadId | | 4 | Application's main thread id as unsigned little-endian 32bit integer for Ok command |

#### *4.4.5.8 Start document*

Start an application to open specified document.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x14<br>0x15 | 1 | StartDoc, StartDoc_u |
| DocumentNameLength | | 1 | Document name length in characters |
| DocumentName | | variable | Document full name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

Ok, NotFound or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0<br>0xF1 | 1 | Ok, NotFound |
| ThreadId | | 4 | Application's main thread id as unsigned little-endian 32bit integer for Ok command |

#### *4.4.5.9 Get application status by application name*

Check the status of an application.

The `ApplicationName` parameter can be the application caption, full path to the application file (including drive letter) or just the application filename (with or without the file extension).

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x16<br>0x17 | 1 | StatusApp, StatusApp_u |
| ApplicationNameLength | | 1 | Application name length in characters |
| ApplicationName | | variable | Application name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

NotFound, Running or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF1<br>0xF4 | 1 | NotFound, Running |

#### *4.4.5.10 Get application status document name*

Check the status of application handling specified document.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x18<br>0x19 | 1 | StatusDoc, StatusDoc_u |
| DocumentNameLength | | 1 | Document name length in characters |
| DocumentName | | variable | Document name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

NotFound, Running or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF1<br>0xF4 | 1 | NotFound, Running |

#### *4.4.5.11 Get application status by UID*

Check the status of application defined by its UID.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x1A | 1 | StatusAppUid |

| | | | |
|---|---|---|---|
| ApplicationUid | | 4 | Application UID as signed little-endian 32bit integer |

**Return**

NotFound, Running or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF1<br>0xF4 | 1 | NotFound, Running |

### *4.4.5.12 Stop application by application name*

Stop application defined by the `ApplicationName` parameter.

The `ApplicationName` parameter can be the application caption, full path to the application file (including drive letter) or just the application filename (with or without the file extension).

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x1C<br>0x1D | 1 | StopApp, StopApp_u |
| ApplicationNameLength | | 1 | Application name length in characters |
| ApplicationName | | variable | Application name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

Ok, NotFound or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0<br>0xF1 | 1 | Ok, NotFound |

### *4.4.5.13 Stop application by document name*

Stop application handling specified document.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x1E<br>0x1F | 1 | StopDoc, StopDoc_u |
| DocumentNameLength | | 1 | Document name length in characters |
| DocumentName | | variable | Document name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

Ok, NotFound or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0<br>0xF1 | 1 | Ok, NotFound |

### *4.4.5.14 Stop application by UID*

Stop application defined by its UID.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x20 | 1 | StopAppUid |
| ApplicationUid | | 4 | Application UID as signed little-endian 32bit integer |

**Return**

Ok, NotFound or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0<br>0xF1 | 1 | Ok, NotFound |

### *4.4.5.15 List running applications*

List the applications currently running.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x24<br>0x25 | 1 | ListApps, ListApps_u |
| IncludeHidden | 0x00,<br>0x01 | 1 | Whether the response should contain applications marked as hidden. |
| IncludeSystem | 0x00,<br>0x01 | 1 | Whether the response should contain applications marked as system applications. |

**Return**

Ok or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0 | 1 | Ok |
| NumberOfApps | | 2 | The number of applications contained in the following list. Encoded as little-endian unsigned 16bit integer. |
| AppUID | | 4 | Application UID as unsigned little-endian |

| | | | |
|---|---|---|---|
| | | | 32bit integer. |
| CaptionNameLength | | 1 | Length of the application caption in characters. |
| CaptionName | | variable | The application caption. Encoded either as 8-bit or 16-bit text depending on the command variant used. |
| DocumentNameLength | | 1 | Length of the document name in characters. |
| DocumentName | | variable | The name of the document that the application is currently handling. Encoded either as 8-bit or 16-bit text depending on the command variant used. |
| Hidden | 0x00, 0x01 | 1 | Flag indicating is the application marked as hidden. |
| System | 0x00, 0x01 | 1 | Flag indicating is the application marked as system application. |
| Ready | 0x00, 0x01 | 1 | Flag indicating is the application ready. |
| Busy | 0x00, 0x01 | 1 | Flag indicating is the application busy. |
| Closable | 0x00, 0x01 | 1 | Flag indicating is the application responding to shutdown events. |

The AppUid, CaptionNameLength, CaptionName, DocumentNameLength, DocumentName, Hidden, System, Ready, Busy and Closable fields are repeated for each application returned by the list command. If there are no applications found, the NumberOfApps is defined as zero

#### *4.4.5.16 List processes*

List processes. An optional search pattern may be given.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0C 0x0D | 1 | ListProcesses, ListProcesses_u |
| MatchPatternLength | | 1 | The match pattern length in characters. |
| MatchPattern | | Variable | The match pattern for the listed processes either encoded as 8-bit text or as UCS-2 for Unicode variant command.<br>This parameter is optional. All available processes are listed, if this parameter is omitted. |

**Return**

Ok or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0 | 1 | Ok |
| NumberOfProcesses | | 2 | The number of listed processes as little-endian unsigned 16bit integer. |
| ProcessId | | 4 | Process id as unsigned little-endian 32bit integer. |
| Status | 0xF4<br>0xF5<br>0xF6 | 1 | Running<br>Killed<br>Panic |
| ProcessNameLength | | 1 | The process name length in characters. |
| ProcessName | | Variable | The process name encoded as 8-bit text. |

The ProcessId, Status, ProcessNameLength and ProcessName fields are repeated for each process returned by list command. If there are no processes found by the defined match pattern, the NumberOfProcesses is defined as zero.

#### *4.4.5.17 Start process for exit code*

Starts a process and stores its process handle for process exit code queries with GetProcessExitCode command.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0E<br>0x07 | 1 | StartProcessRetVal,<br>StartProcessRetVal_u |
| ProgramNameLength | | 1 | Program name length in characters |
| ProgramName | | Variable | Program full name either encoded as 8-bit text or as UCS-2 for Unicode variant command |
| CommandLineArgumentLength | | 1 | Command line argument string length in characters |
| CommandLineArgument | | Variable | Command line argument string either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

Ok, NotFound or error message (see 4.3.3).

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0<br>0xF1 | 1 | Ok, NotFound |
| ProcessId | | 4 | Process id as unsigned little-endian 32bit integer for Ok command |

### 4.4.5.18 Get process exit code

Gets the status and exit code of the process started with StartProcessRetVal command.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0B | 1 | GetProcessExitCode |
| ProcessId | | 4 | Process Id as unsigned little-endian 32bit integer |

**Return**

Ok or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0<br>0xF1 | 1 | Ok, NotFound |
| ProcessStatus | 0 – 3 | 1 | 0 = Killed<br>1 = Terminated<br>2 = Panic<br>3 = Running |
| ExitReason | | 4 | Exit code or panic code depending on process status. Zero if process is still running. |
| ExitCategoryLength | >=0 | 1 | The length of the exit category field |
| ExitCategory | | >= 0 | The name of the category associated with the end of the process.<br>If the process has panicked, the category name is the panic category name, for example KERN-EXEC.<br>If the process has ended as a result of a call to Kill(), then the category name is Kill. If the process has ended as a result of a call to Terminate(), then the category name is Terminate.<br>If the process is still running, the category name is empty. |

### 4.4.5.19 Install software

Install the software from defined installation package according to given options.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x30<br>0x31 | 1 | Install, Install_u |
| InstallPackagePathLength | >=0 | 1 | Install package path length in characters |
| InstallPackagePath | | Variable | Install package path either encoded as 8-bit text or as UCS-2 for Unicode variant |

| | | | |
|---|---|---|---|
| | | | command |
| Upgrade | 0, 1 | 1 | Whether upgrading is allowed |
| OptionalItems | 0,1 | 1 | Whether installation is allowed if the installation package contains optional items |
| OCSP | 0,1 | 1 | Whether OCSP procedure should be done |
| IgnoreOCSPWarnings | 0,1 | 1 | Whether OCSP warnings are ignored |
| Untrusted | 0,1 | 1 | Whether untrusted packages may be installed |
| PackageInfo | 0,1 | 1 | Whether installation is allowed to continue if package info should be displayed |
| Capabilities | 0,1 | 1 | Whether user capabilities will be granted |
| KillApp | 0,1 | 1 | Whether running applications will be killed if needed |
| Overwrite | 0,1 | 1 | Whether existing files will be overwritten if needed |
| Download | 0,1 | 1 | Whether download will be performed, if needed by installation |
| DownloadUsernameLength | >=0 | 1 | The username length in characters |
| DownloadUsername | | Variable | The username to be provided to web page if needed during download either encoded as 8-bit text or as UCS-2 for Unicode variant command |
| DownloadPasswordLength | >=0 | 1 | The password length in characters |
| DownloadPassword | | Variable | The password to be provided to web page if needed during download either encoded as 8-bit text or as UCS-2 for Unicode variant command |
| Drive | | 1 | The target drive to install (e.g. C, E).<br>! character stands for any drive having enough space |
| Language | | 1 | The target language according to the TLanguage enumeration defined in e32const.h |
| UsePhoneLanguage | 0,1 | 1 | If set to TRUE, the preceding Language parameter is ignored and phone's default language is used instead |
| UpgradeData | 0,1 | 1 | Whether the data of the application should be upgraded as part of the package upgrade |

**Return**

Ok, NotFound or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0 | 1 | Ok |

### 4.4.5.20 Uninstall software

Uninstall the software with defined UID.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x32 | 1 | UnInstall |
| ApplicationUid | | 4 | Application UID as signed little-endian 32bit integer |
| KillApp | 0,1 | 1 | Whether running applications will be killed if needed |
| BreakDependency | 0,1 | 1 | Whether it is acceptable to continue uninstallation if it will cause a dependency break |
| MimeType | 0…7 | 1 | Mime type of the package to be uninstalled according to the following enumeration:<br>0 = Sisx Mime Type,<br>1 = Sis Mime Type,<br>2 = Pip Mime Type,<br>3 = Jad Mime Type,<br>4 = Jar Mime Type,<br>5 = Java Mime Type,<br>6 = Jarx Mime Type<br>7= Other MIME-Types (*.*) |

**Return**

Ok, NotFound or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0 | 1 | Ok, |

### 4.4.5.21 Uninstall software by name

Uninstall software with defined install package name.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x33,<br>0x34 | 1 | 0x33 = UnInstallName_u,<br>0x34 = UnInstallName |
| PackageNameLength | | 1 | The length of the following name field in characters. |
| PackageName | | Variable | Package name either encoded as 8-bit text or as UCS-2 for Unicode variant command. |
| KillApp | 0,1 | 1 | Whether running applications will be killed if needed. |
| BreakDependency | 0,1 | 1 | Whether it is acceptable to continue uninstallation if it will cause a dependency break. |
| MimeType | 0…7 | 1 | Mime type of the package to be uninstalled |

| | | | |
|---|---|---|---|
| | | | according to the following enumeration:<br>0 = Sisx Mime Type,<br>1 = Sis Mime Type,<br>2 = Pip Mime Type,<br>3 = Jad Mime Type,<br>4 = Jar Mime Type,<br>5 = Java Mime Type,<br>6 = Jarx Mime Type<br>7= Other MIME-Types (*.*) |

**Return**

Ok, NotFound or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0xF0 | 1 | Ok |

### *4.4.6 FTP SERVICE (FTP)*

The FTP service provides functions to remotely manage phone files. It includes files sending/retrieving/deleting, and directory listing/creating/deleting. The FTP service plugin uses the Priorityflag (see Table 1) to organise control and data channels. Data channel (messages with normal priority) is used for actual file transfer and control channel (higher priority message) used for requests. In reply to any command except CANCEL (see chapter 0), an error message (see 4.3.3) with plug-in error code -16 can be sent. It happens when the FTP service is currently transferring file and is not able to process another command.

The following tables list all supported commands, endings "_u" means the variant command that accepts or reply with directory and file names in Unicode.

**Table 13: FTP commands**

| CommandName | CommandCode |
|---|---|
| FtpSTOR | 0x02 |
| FtpSTOR_u | 0x03 |
| FtpRETR | 0x04 |
| FtpRETR_u | 0x05 |
| FtpLIST | 0x06 |
| FtpLIST_u | 0x07 |
| FtpMKD | 0x08 |
| FtpMKD_u | 0x09 |
| FtpRMD | 0x0A |
| FtpRMD_u | 0x0B |
| FtpDELE | 0x0C |
| FtpDELE_u | 0x0D |

| | |
|---|---|
| FtpCANCEL | 0x0E |
| FtpFILESIZE | 0x0F |
| FtpLISTDIR | 0x10 |
| FtpLISTDIR_u | 0x11 |
| FtpLISTSIZES | 0x12 |
| FtpLISTSIZES_u | 0x13 |
| FtpSETFORCE | 0x20 |
| FtpCHECKSUM | 0x30 |
| FtpCHECKSUM_u | 0x31 |
| FtpOK | 0xF0 |

For OK response, message body contains only FtpOK command code.

#### *4.4.6.1 Upload file*

Copy file to a phone. After the request command is sent, framework will reply with an error message if the file can not be accepted or FtpOk if it is accepted. If FtpOk message sent, HtiFramework expects file data in messages in the data channel. Data transfer can be cancelled by sending the CANCEL command (see chapter 0). When all data are received, FtpOk is send back again.

**Note:** Uploading files to TCB directories (/sys and /resource) requires HTI to have TCB capability. If you have a production device and a developer certificate that does not allow TCB capability then the only way to get files to TCB folders is to create a signed SIS package of the files and put it through the installer. For automatic SIS install, you can use the Install command of the HTI application service.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x02<br>0x03 | 1 | FtpSTOR, FtpSTOR_u |
| FileSize | | 4 | File size as little-endian 32bit integer |
| FileNameLength | | 1 | FileName length in characters |
| FileName | | Variable | File name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

FtpOK or error message (see 4.3.3)

#### *4.4.6.2 Download file*

Retrieve a file from phone. In response to the RETR command, FTP service will send control message FILESIZE that specifies the size of the file to be transferred. After this, file data are sent in the data channel. Data transfer can be canceled by sending the CANCEL command (see chapter 0).

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x04<br>0x05 | 1 | `FtpRETR, FtpRETR_u` |
| FileNameLength | | 1 | FileName length in characters |
| FileName | | Variable | File name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0F | 1 | `FtpFILESIZE` |
| FileSize | | 4 | File size as little-endian 32bit integer |

#### *4.4.6.3 Cancel file transfer*

This command is used to cancel current data transfer.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0E | 1 | `FtpCANCEL` |

**Return**

FtpOK or error message (see 4.3.3)

#### *4.4.6.4 List files*

Returns the list of files located inside a specified directory. The response contains the list of files where each file name is represented by file name length in the first byte with the file name following.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x06<br>0x07 | 1 | `FtpLIST, FtpLIST_u` |
| DirectoryNameLength | | 1 | DirectoryName length in characters |
| DirectoryName | | Variable | Directory name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

The file list or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| FileNameLength | | 1 | FileName length in characters |
| FileName | | Variable | File name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

### *4.4.6.5 List directories*

Returns the list of directories located inside a specified directory. The response contains the list of directories where each directory name is represented by name length in the first byte with the directory name following.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x10<br>0x11 | 1 | `FtpLISTDIR, FtpLISTDIR_u` |
| DirectoryNameLength | | 1 | DirectoryName length in characters |
| DirectoryName | | Variable | Directory name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

The directory list or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| DirNameLength | | 1 | DirName length in characters |
| DirName | | Variable | Directory name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

### *4.4.6.6 List files with sizes*

Returns the list of files located inside a specified directory. The response contains the list of files. Each file entry consists of file name length byte, the file name string and the size of the file in bytes.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x12<br>0x13 | 1 | `FtpLISTSIZES, FtpLISTSIZES_u` |
| DirectoryNameLength | | 1 | DirectoryName length in characters |
| DirectoryName | | Variable | Directory name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

The file list or error message (see 4.3.3)

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| FileNameLength | | 1 | FileName length in characters |
| FileName | | Variable | File name either encoded as 8-bit text or as UCS-2 for Unicode variant command |
| FileSize | | 4 | File size as little-endian 32bit unsigned integer |

#### *4.4.6.7 Create directory*

Creates specified directory.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x08<br>0x09 | 1 | `FtpMKD, FtpMKD_u` |
| DirectoryNameLength | | 1 | DirectoryName length in characters |
| DirectoryName | | Variable | Directory name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

FtpOK or error message (see 4.3.3)

#### *4.4.6.8 Delete directory*

Deletes specified directory.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0A<br>0x0B | 1 | `FtpRMD, FtpRMD_u` |
| DirectoryNameLength | | 1 | DirectoryName length in characters |
| DirectoryName | | Variable | Directory name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

FtpOK or error message (see 4.3.3)

#### *4.4.6.9 Delete file*

Deletes specified file.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0C<br>0x0D | 1 | FtpDELE, FtpDELE_u |
| FileNameLength | | 1 | FileName length in characters |
| FileName | | Variable | File name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

FtpOK or error message (see 4.3.3)

### *4.4.6.10 Set forced operations*

Sets forced file operations on or off.

This functionality can be used if there is a need to download, delete or overwrite a file that is locked by some system application and cannot be accessed. When forced operations is set on, FtpRETR, FtpSTOR and FtpDELE commands will try the forced operation on the file if it otherwise fails.

In practice the forced operation means faking a backup/restore operation, which will shutdown GUI applications and make the system applications to release the file locks on the system files.

Note: that this is quite heavy operation and changes the state of device and applications, so this should be used only if absolutely necessary.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x20 | 1 | FtpSETFORCE |
| State | 0, 1 | 1 | Sets the state of forced file operations. 0 = forced operations off, 1 = forced operations on |

**Return**

FtpOK or error message (see 4.3.3)

### *4.4.6.11 File checksum*

Calculate a checksum for a file. This can be used for example to compare equality of files or verifying the integrity of a file after file transfer operations.

The following table lists the currently supported algorithms, their ID numbers (ID to be used in the command message) and the length of the calculated checksum.

| Algorithm | AlgorithmId | ChecksumLength |
|---|---|---|
| MD5 | 0x01 | 16 bytes |

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x30<br>0x31 | 1 | FtpCHECKSUM, FtpCHECKSUM_u |
| AlgorithmId | 0x01 | 1 | The ID number of algorithm to use in checksum calculation |
| FileNameLength | | 1 | FileName length in characters |
| FileName | | Variable | File name either encoded as 8-bit text or as UCS-2 for Unicode variant command |

**Return**

The calculated checksum bytes (see length from the supported algorithms table above) or error message (see 4.3.3)

#### *4.4.6.12 Format*

Formats a drive (memory card).

Will not format the drive if HTI client is running from that drive.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x40 | 1 | FtpFORMAT |
| Drive | | 1 | The ASCII character code of the drive letter representing the drive to be formatted. For example 0x45 for drive E. |
| Mode | 0x00,<br>0x01 | 1 | The format mode: 0 = full format, 1 = quick format |

**Return**

FtpOK or error message (see 4.3.3)

### *4.4.7 SYSINFO SERVICE*

This service allows to retrieve different information about the phone and to control memory, lights and some system settings.

The following commands are supported:

**Table 14: SysInfo commands**

| CommandName | CommandCode |
|---|---|
| HAL | 0x01 |
| IMEI | 0x02 |

| | |
|---|---|
| SWVersion | 0x03 |
| LangVersion | 0x04 |
| SWLangVersion | 0x05 |
| UserAgentString | 0x06 |
| GetFreeRAM | 0x07 |
| GetUsedRAM | 0x08 |
| GetTotalRAM | 0x09 |
| EatRAM | 0x0A |
| ReleaseRAM | 0x0B |
| GetFreeDiskSpace | 0x0C |
| GetUsedDiskSpace | 0x0D |
| GetTotalDiskSpace | 0x0E |
| EatDiskSpace | 0x0F |
| ReleaseDiskSpace | 0x10 |
| SetHomeTime | 0x20 |
| GetHomeTime | 0x21 |
| LightStatus | 0x30 |
| LightOn | 0x31 |
| LightOff | 0x32 |
| LightBlink | 0x33 |
| LightRelease | 0x3A |
| ScreenSaverDisable | 0x40 |
| ScreenSaverEnable | 0x41 |
| ScreenSaverTimeout | 0x42 |
| GetNetworkMode | 0x50 |
| SetNetworkMode | 0x51 |
| SetNetworkModeNoReboot | 0x52 |
| SetHsdpa | 0x53 |
| IrActivate | 0x5A |
| BtPower | 0x5B |
| BtSettings | 0x5C |
| KeyLockToggle | 0x60 |
| AutoKeyGuardTime | 0x61 |
| EmptyDrmRightsDb | 0x65 |
| BatteryStatus | 0x70 |

#### 4.4.7.1 HAL

This command accepts a 32bit integer argument corresponding to HALData::TAttribute enumeration. The return value is a 32bit integer value, which depends on the argument.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x01 | 1 | HAL |
| AttributeValue | | 4 | Value from HALData::TAttribute enumeration |

**Return**

Response code as 32-bit little-endian integer or error message (see 4.3.3)

#### 4.4.7.2 IMEI

Returns IMEI code as a 8-bit encoded string.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x02 | 1 | IMEI |

**Return**

IMEI as 8-bit string or error message (see 4.3.3)

#### 4.4.7.3 SWVersion

Returns SW version as Unicode UCS-2 encoded string.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x03 | 1 | SWVersion |

**Return**

SWVersion as Unicode UCS-2 string or error message (see 4.3.3)

#### 4.4.7.4 LangVersion

Returns the version of the currently installed language package as Unicode UCS-2 encoded string.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x04 | 1 | LangVersion |

**Return**

LangVersion as Unicode UCS-2 string or error message (see 4.3.3)

#### *4.4.7.5 SWLangVersion*

Returns the SW version with which the currently installed Language package is compatible with as Unicode UCS-2 encoded string.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x05 | 1 | SWLangVersion |

**Return**

SWLangVersion as Unicode UCS-2 string or error message (see 4.3.3)

#### *4.4.7.6 UserAgentString*

Returns the device specific user agent string (used e.g. in HTTP headers) as 8-bit encoded string.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x06 | 1 | UserAgentString |

**Return**

UserAgentString as 8-bit string or error message (see 4.3.3)

#### *4.4.7.7 GetFreeRAM*

Return the amount of free RAM.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x07 | 1 | GetFreeRAM |

**Return**

The amount of free RAM as 32-bit little-endian integer or error message (see 4.3.3)

### 4.4.7.8 GetUsedRAM

Return the amount of used RAM.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x08 | 1 | GetUsedRAM |

**Return**

The amount of used RAM as 32-bit little-endian integer or error message (see 4.3.3)

### 4.4.7.9 GetTotalRAM

Return the amount of total RAM.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x09 | 1 | GetTotalRAM |

**Return**

The amount of total RAM as 32-bit little-endian integer or error message (see 4.3.3)

### 4.4.7.10 EatRAM

Eat memory so that the amount of free RAM is set to defined level.

Note: The maximum amount of RAM that HTI can reserve is by default 1048576 bytes (one megabyte). The MaxHeapSize variable in HTI's configuration file can be increased to allow HTI consume more memory.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0A | 1 | EatRAM |
| RAMToBeLeftOver | | 4 | The amount of RAM to be left over as 32-bit little-endian integer |

**Return**

The amount of free RAM after it has been eaten as 32-bit little-endian integer or error message (see 4.3.3)

### 4.4.7.11 ReleaseRAM

Release the previously eaten RAM.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0B | 1 | `ReleaseRAM` |

**Return**

The amount of free RAM after the previously eaten RAM has been released as 32-bit little-endian integer or error message (see 4.3.3).

#### *4.4.7.12 GetFreeDiskSpace*

Return the amount of free disk space on the defined volume.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0C | 1 | `GetFreeDiskSpace` |
| DriveLetter | A-Z | 1 | The drive letter of volume whose disk space is queried |

**Return**

The amount of free disk space as 64-bit little-endian integer or error message (see 4.3.3)

#### *4.4.7.13 GetUsedDiskSpace*

Return the amount of used disk space on the defined volume.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0D | 1 | `GetUsedDiskSpace` |
| DriveLetter | A-Z | 1 | The drive letter of volume whose disk space is queried |

**Return**

The amount of used disk space as 64-bit little-endian integer or error message (see 4.3.3)

#### *4.4.7.14 GetTotalDiskSpace*

Return the amount of total disk space on the defined volume.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0E | 1 | `GetTotalDiskSpace` |
| DriveLetter | A-Z | 1 | The drive letter of volume whose disk |

| | | | |
|---|---|---|---|
| | | | space is queried |

**Return**

The amount of total disk space as 64-bit little-endian integer or error message (see 4.3.3)

#### *4.4.7.15 EatDiskSpace*

Eat disk space on the specified volume so that the amount of free disk space is set to defined level.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x0F | 1 | EatDiskSpace |
| DriveLetter | A-Z | 1 | The drive letter of volume whose disk space is eaten |
| DiskSpaceToBeLeftOver | | 8 | The amount of disk space to be left over as 64-bit little-endian integer |

**Return**

The amount of free disk space after it has been eaten as 64-bit little-endian integer or error message (see 4.3.3)

#### *4.4.7.16 ReleaseDiskSpace*

Release previously eaten disk space.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x10 | 1 | ReleaseDiskSpace |
| DriveLetter | A-Z | 1 | The drive letter of volume whose disk space is released |

**Return**

The amount of free disk space after it has been released as 64-bit little-endian integer or error message (see 4.3.3)

#### *4.4.7.17 SetHomeTime*

Sets the date and time of the device to the specified value.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x20 | 1 | SetHomeTime |
| Year | | 2 | The year value to set. Encoded as unsigned |

| | | | |
|---|---|---|---|
| | | | little-endian 16bit integer. |
| Month | 0x01 - 0x0C | 1 | The month value to set as integer. |
| Day | 0x01 - 0x1F | 1 | The day value to set as integer. |
| Hour | 0x00 - 0x17 | 1 | The hour value to set as integer. |
| Minute | 0x00 - 0x3B | 1 | The minute value to set as integer. |
| Second | 0x00 - 0x3B | 1 | The second value to set as integer. |

**Return**

Byte 0x00 if setting the time succeeded.

Error message (see 3.3) if the given time is not valid or if setting the time fails.

### *4.4.7.18 GetHomeTime*

Gets the current home time of the device (the time being displayed to the user).

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x21 | 1 | `GetHomeTime` |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Year | | 2 | The year value. Encoded as unsigned little-endian 16bit integer. |
| Month | 0x01 - 0x0C | 1 | The month value as integer. |
| Day | 0x01 - 0x1F | 1 | The day value as integer. |
| Hour | 0x00 - 0x17 | 1 | The hour value as integer. |
| Minute | 0x00 - 0x3B | 1 | The minute value as integer. |
| Second | 0x00 - 0x3B | 1 | The second value as integer. |

### *4.4.7.19 DateTimeFormat*

Sets the date and time format settings.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x22 | 1 | DateTimeFormat |
| DateFormat | 0x00 – 0x02 | 1 | The date presentation format:<br>0 = American (mm/dd/yyyy)<br>1 = European (dd/mm/yyyy)<br>2 = Japanese (yyyy/mm/dd) |
| DateSeparatorCharacter | 0x2D,<br>0x2E,<br>0x2F,<br>0x3A, | 1 | The ASCII code of the character to be used as separator between day, month and year. |
| TimeFormat | 0x00,<br>0x01 | 1 | Whether to use 12 or 24 hour time format. 0 = 12 hour, 1 = 24 hour |
| TimeSeparatorCharacter | 0x2E,<br>0x3A | 1 | The ASCII code of the character to be used as separator between hours and minutes. |
| ClockFormat | 0x00,<br>0x01 | 1 | Whether to show analog or digital clock. 0 = analog, 1 = digital |

**Return**

Byte 0x00 if successful. Error message (see 3.3) if failed.

### *4.4.7.20 About Light command parameters*

The following table explains some common characteristics related to the following Light-command parameters.

| Parameter | Description |
|---|---|
| LightTarget | The LightTarget is defined as a bitmask.<br>Some common values are: 1 = Primary display, 2 = Primary keyboard, 4 = Secondary display, 8 = Secondary keyboard. These can be combined as bitmask, so for example 3 = Primary display and keyboard, 12 = Secondary display and keyboard, and so on.<br>The LightStatus command can only handle one target, not combinations.<br>Note that all devices do not support all targets and specifying non-existing targets can result to “not supported” errors. |
| Duration | The duration is defined as milliseconds and since it is 2-bytes, the maximum value is 65535 milliseconds. Giving value 0 as duration means infinite duration.<br>Note that if duration is something else than 0, other applications controlling the lights (mainly SysAp that is controlling the lights based on user activity/inactivity) may interrupt the given command during its duration.<br>An infinite duration command can be stopped by calling some other light control command. |
| Intensity | Intensity is defined as percentage value (1 - 100). Defining intensity value 0 results in using the device’s default intensity.<br>All devices might not support intensity definition. In that case the device will behave in its default fashion. |
| FadeIn / FadeOut | Defines whether lights are turned on/off instantly of smoothly fade in/out.<br>All devices might not support fade in/out. In that case the device will behave in its default fashion. |

| | |
|---|---|
| OnDuration / OffDuration | The duration is defined as milliseconds and since it is 2-bytes, the maximum value is 65535 milliseconds. Giving value 0 resutls in using the device default values.<br>If either the on or off value is defined as 0 then both must be 0.<br>All devices might not support variable blink times. In that case the device default values are used. |

#### *4.4.7.21 LightStatus*

Returns the current status of one light target.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x30 | 1 | `LightStatus` |
| LightTarget | 0x01, 0x02, 0x04, 0x08 | 1 | The light target whose status to retrieve. |

**Return**

LightStatus as one byte integer: 1 = Light is on, 2 = Light is off, 3 = Light is blinking.

Error message (see 3.3) if there is a failure in getting the status.

#### *4.4.7.22 LightOn*

Turns on the specified light target with given duration and intensity.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x31 | 1 | `LightOn` |
| LightTarget | | 1 | The light target to turn on. |
| Duration | 0x0000 - 0xFFFF | 2 | How long to keep the light on. Specified in milliseconds as unsigned little-endian 16bit integer. |
| Intensity | 0x00 - 0x64 | 1 | The light intensity (brightness) to use defined as percentage value. |
| FadeIn | 0x00, 0x01 | 1 | Whether to turn on light directly to defined intensity or with fade effect.<br>0 = no fade, any other value = use fade |

**Return**

Byte 0x00 if successful. Error message (see 3.3) if failed.

### 4.4.7.23 LightOff

Turn off the specified light target for given duration.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x32 | 1 | LightOff |
| LightTarget | | 1 | The light target to turn off. |
| Duration | 0x0000 - 0xFFFF | 2 | How long to keep the light off. Specified in milliseconds as unsigned little-endian 16bit integer. |
| FadeOut | 0x00, 0x01 | 1 | Whether to turn off light immediately or with fade effect.<br>0 = no fade, any other value = use fade |

**Return**

Byte 0x00 if successful. Error message (see 3.3) if failed.

### 4.4.7.24 LightBlink

Blink the specified light target.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x33 | 1 | LightBlink |
| LightTarget | | 1 | The light target to blink. |
| Duration | 0x0000 - 0xFFFF | 2 | How long to keep the light blinking. Specified in milliseconds as unsigned little-endian 16bit integer. |
| OnDuration | 0x0000 - 0xFFFF | 2 | How long to keep the light on in each blink cycle. Specified in milliseconds as unsigned little-endian 16bit integer. |
| OffDuration | 0x0000 - 0xFFFF | 2 | How long to keep the light off in each blink cycle. Specified in milliseconds as unsigned little-endian 16bit integer. |
| Intensity | 0x00 - 0x64 | 1 | The light intensity (brightness) to use defined as percentage value. |

**Return**

Byte 0x00 if successful. Error message (see 3.3) if failed.

### 4.4.7.25 LightRelease

Releases control over all light targets. The effect of previously given light command is cancelled.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x3A | 1 | LightRelease |

**Return**

Byte 0x00 if successful. Error message (see 3.3) if failed.

### *4.4.7.26 ScreenSaverDisable*

Disable screen saver for being activated.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x40 | 1 | ScreenSaverDisable |

**Return**

Byte 0x00 if successful. Error message (see 3.3) if failed.

### *4.4.7.27 ScreenSaverEnable*

Cancels a previously given ScreenSaverDisable request. Note that other applications may also have disabled the screen saver so it is not necessarily enabled after this command. This command just cancels the HTI-given disable request.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x41 | 1 | ScreenSaverEnable |

**Return**

Byte 0x00 if successful. Error message (see 3.3) if failed.

### *4.4.7.28 ScreenSaverTimeout*

Sets the screen saver timeout value.

Note that from the phone user interface the screen saver timeout can be set between 5 – 90 seconds with 5 second intervals. Although HTI allows any value between 5 – 255, it is recommended to use only values that can be set also from the phone settings UI.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x42 | 1 | ScreenSaverTimeout |
| Timeout | 0x05 – | 1 | The screen saver timeout in seconds. |

| | | | |
|---|---|---|---|
| | 0xFF | | |

**Return**

Byte 0x00 if successful. Error message (see 3.3) if failed.

### *4.4.7.29 GetNetworkMode*

Get the current network mode of the device.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x50 | 1 | GetNetworkMode |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x50 | 1 | GetNetworkMode |
| NetworkMode | | 4 | Bitmask of currently active network mode as unsigned little-endian 32bit integer.<br>Gsm = 0x01<br>Umts = 0x02<br>Dual = 0x04 |

Error message (see 3.3) if failed.

### *4.4.7.30 SetNetworkMode*

Set the current network mode of the device. This command will also reboot the device.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x51 | 1 | SetNetworkMode |
| NetworkMode | | 4 | Bitmask of network mode to set as unsigned little-endian 32bit integer.<br>Gsm = 0x01<br>Umts = 0x02<br>Dual = 0x04 |

**Return**

If the command is successful there is no return message because HTI is stopped and the device rebooted. However if an error occurs standard error message is returned (see 3.3).

#### 4.4.7.31 SetNetworkModeNoReboot

Sets the current network mode of the device. This command will not reboot the device.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x52 | 1 | `SetNetworkModeNoReboot` |
| NetworkMode | | 4 | Bitmask of network mode to set as unsigned little-endian 32bit integer.<br>Gsm = 0x01<br>Umts = 0x02<br>Dual = 0x04 |

**Return**

Byte 0x00 if successful. Error message (see 3.3) if failed.

#### 4.4.7.32 SetHsdpa

Enables or disables the “High speed packet access”.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x53 | 1 | `SetHsdpa` |
| Enable/Disable | 0x00 or 0x01 | 1 | 0 = Disable HSDPA, 1 = Enable HSDPA |

**Return**

Byte 0x00 if status was changed successfully, byte 0x01 if it was already in requested state and no change was made. Error message (see 3.3) if failed

#### 4.4.7.33 IrActivate

Activates the infrared listening.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x5A | 1 | `IrActivate` |

**Return**

Byte 0x00 if IR activated, byte 0x01 if IR was already active. Error message (see 3.3) if failed.

#### 4.4.7.34 BtPower

Controls Bluetooth power state.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x5B | 1 | `BtPower` |
| BtPowerState | 0x00, 0x01 | 1 | 0 = turn Bluetooth off, 1 = turn Bluetooth on |
| Force | 0x00, 0x01 | 1 | Force power state change: 0 = do not force, 1 = force.<br>Force must be used if setting BT on when in Offline mode or setting BT off when there are active BT connections. |

**Return**

Byte 0x00 if power state change was successful, byte 0x01 if Bluetooth was already in requested state. Error message (see 3.3) if failed

### *4.4.7.35 BtSettings*

Controls Bluetooth settings.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x5C | 1 | `BtSettings` |
| Discoverability | 0x00, 0x01 | 1 | 0 = hidden, 1 = discoverable |
| EnableSAP | 0x00, 0x01 | 1 | 0 = SIM access profile disabled, 1 = SIM access profile enabled |
| BtNameLength | 0x00 – 0x1E | 1 | Length of the following Bluetooth name field. Zero means that BT name will not be changed. |
| BtName | | 0 – 30 | Bluetooth name to be set encoded as 8-bit text. |

**Return**

Byte 0x00 if successful. Error message (see 3.3) if failed.

### *4.4.7.36 BtDeletePairings*

Delete Bluetooth pairings.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x5D | 1 | `BtDeletePairings` |
| CloseConnections | 0x00, 0x01 | 1 | 0 = do not delete pairing if device has active connection, 1 = if device has active connection, close it and delete pairing |
| BtNameLength | 0x00 – | 1 | Length of the following Bluetooth name field. |

| | | | |
|---|---|---|---|
| | 0xF8 | | Zero means that all BT pairings will be deleted. |
| BtName | | 0 – 248 | Exact name or match pattern indicating which BT device pairings to delete. Wildcard characters ? and * can be used to specify a match pattern. |

**Return**

The number of device pairings deleted (as one byte) if successful. Error message (see 3.3) if any deletion failed.

### *4.4.7.37 KeyLockToggle*

This command turns the device keypad locking on or off. It can also be controlled whether the information dialog about key lock status change is showed or not.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x60 | 1 | KeyLockToggle |
| RequestedStatus | 0x00, 0x01 | 1 | 0 = unlock keys, 1 = lock keys |
| ShowInfoNote | 0x00, 0x01 | 1 | 0 = do not show note, 1 = show note |

**Return**

Byte 0x00 if key lock state was changed successfully. Byte 0x01 if key lock was already in requested state. Error message (see 3.3) if failed

### *4.4.7.38 AutoKeyGuardTime*

This command controls the time setting of automatic keypad lock (key guard). If value is set to zero, automatic keypad locking is disabled.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x61 | 1 | AutoKeyGuardTime |
| TimeValue | 0x0000 - 0x0E10 | 2 | The auto key lock time to set in seconds. Maximum value is 3600 seconds (1 hour). Zero value disables automatic key locking. |

**Return**

Byte 0x00 if successful. Error message (see 3.3) if failed.

#### 4.4.7.39 EmptyDrmRightsDb

This command will erase existing DRM rights objects from the DRM rights database.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x65 | 1 | EmptyDrmRightsDb |

**Return**

Byte 0x00 if successful. Error message (see 3.3) if failed.

#### 4.4.7.40 BatteryStatus

This command returns the current battery charge level and the current charging status.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x70 | 1 | BatteryStatus |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| BatteryBars | 0x00 – 0x07 | 1 | The current charge level of the battery as in how many bars are displayed by the battery status indicator. |
| ChargingStatus | 0x00 – 0x05 | 1 | The current charging status:<br>0 = charger not connected<br>1 = device is charging<br>2 = charger is connected, device is not charging<br>3 = charging almost completed<br>4 = charging completed<br>5 = charging continued after brief interruption |

Error message (see 3.3) if failed.

#### 4.4.7.41 SignalStrength

This command returns the current network signal strength.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x71 | 1 | SignalStrength |

**Return**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| SignalBars | 0x00 – 0x07 | 1 | The current signal strength as in how many bars are displayed by the signal strength indicator. |

Error message (see 3.3) if failed.

#### *4.4.7.42 UpdateMediaGallery*

This command updates the contents of the Media Gallery. This command should be used if media files are transferred to media folders. Media Gallery application does not detect and show the new files unless it is explicitly told to scan for new file(s).

It is not mandatory to give the path to the file (PathLength can be set to zero and omit FilePath). If path is not given Media Gallery will scan all new files. However, if path to the new file is known it might be faster to update just that file.

**Syntax**

| Field | Value | Size(bytes) | Description |
|---|---|---|---|
| Command | 0x7A | 1 | `UpdateMediaGallery` |
| PathLength | 0x00 – 0xFF | 1 | The length of the following file path in characters. |
| FilePath | | 0x00 – 0xFF | The full path to the file that was uploaded |

**Return**

Byte 0x00 if successful. Error message (see 3.3) if failed

### *4.4.8 AUDIO CONTROL SERVICE*

This service allows playing of audio tones and samples in the device.

The service supports the following commands:

| Command/ Message Name | Command / Message Code |
|---|---|
| `ListAudioFiles` | 0x01 |
| `PlayFile` | 0x02 |
| `PlayTone` | 0x03 |
| `PlayDTMF` | 0x04 |
| `Stop` | 0x05 |

| | |
|---|---|
| GetDuration | 0x06 |
| GetMaxVolume | 0x07 |
| SetVolume | 0x08 |

Unicode versions of the commands are currently not supported. This means that the service does not support filepaths and filenames that can not be encoded as 8-bit text.

The format for each command and response messages are defined in the following sub-chapters.

In case there is an error when executing the command a framework error message (see 3.3) is returned. In the error message the ServiceErrorCode contains an error code and the first byte of the ServiceErrorDescription is the CommandCode of the command that was processed when the error occured. Following bytes of the ServiceErrorDescription may contain textual description of the error.

#### *4.4.8.1 List Audio Files*

Returns a list of audio files. If the command is given without any parameters it searches for audio files under the default sound directory location in ROM, phone memory and memory card. Also all subdirectories under that directory are searched. If the directory parameter is given then audio files are searched from that directory and its subdirectories.

The command returns files whose MIME type starts with audio/ and files whose MIME type is application/vnd.nokia.ringing-tone. The command can return also files whose format is not supported by the PlayFile command (e.g. playlist files and links to Real Audio streams).

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x01 | 1 | ListAudioFiles |
| PathLength | NO | | 1 | Length of the DirectoryPath parameter in characters |
| DirectoryPath | NO | | variable | Absolute path to the directory to scan for audio files |

**Response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x01 | 1 | ListAudioFiles |
| FileCount | YES | | 2 | Amount of audio files found (16-bit unsigned little-endian integer) |
| PathLength | NO | | 1 | Length of the following FullFilePath in |

| | | | | |
|---|---|---|---|---|
| | | | | characters |
| FullFilePath | NO | | variable | Full path to the first found audio file encoded as 8-bit text |
| … PathLength and FullFilePath repeated as many times as indicated by the FileCount | | | | |

#### *4.4.8.2 Play File*

Starts playback of an audio file if the file format is supported. Supported formats are those supported by the device's Media Server and the Nokia rng ringtone format.

All parameters are mandatory. Setting the StartPosition and EndPosition both to zero will play the complete audio sample. Setting StartPosition and EndPosition is not supported when playing Nokia rng ringtone format file. In case the file is rng ringtone file the StartPosition and EndPosition parameters have no effect.

Response message is sent when the playback ends.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x02 | 1 | PlayFile |
| PathLength | YES | | 1 | Length of the FullFilePath parameter in characters |
| FullFilePath | YES | | variable | Full path to the audio file encoded as 8-bit text. |
| Volume | YES | | 1 | Playback volume as integer. |
| StartPosition | YES | | 4 | The position from where to start the playback measured as microseconds from the beginning of the file. Encoded as unsigned little-endian 32bit integer. |
| EndPosition | YES | | 4 | The position where to end the playback measured as microseconds from the beginning of the file. Encoded as unsigned little-endian 32bit integer.<br>Value must be greater or equal than StartPosition. If value is set greater than the duration of the file the playback ends at the end of file. |
| NumberOfRepeats | YES | | 1 | How many times to repeat the playback of the file. Defined as integer. Zero value means that playback is done once, one means played two times, etc. |
| SilenceBetweenRepeats | YES | | 4 | The pause to keep between repeats defined as microseconds. Encoded as unsigned little-endian 32bit integer. |
| AudioSettings | YES | 0x00 - 0x03 | 1 | The settings to use when playing the file. Possible values are described in the audio settings table below. |

**Response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x02 | 1 | PlayFile |
| OkResponse | YES | OK (8-bit string) | 2 | Response that the playback was completed successfully. |

AudioSettings parameter can have the following values:

| Setting | Value | Description |
|---|---|---|
| `Default` | 0x00 | Play with default settings |
| `General Music` | 0x01 | Play with settings for media players (such as RealOne Player) |
| `Ring Tone Preview` | 0x02 | Play with settings used when previewing ring tones. |

If any other value is specified it will be translated to `Default`.

### *4.4.8.3 Play Tone*

Plays a single sine tone with specified frequency and duration.

Response message is sent when the playback ends.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x03 | 1 | PlayTone |
| Frequency | YES | | 2 | Frequency to play in hertz defined as 16-bit unsigned little-endian integer. |
| Duration | YES | | 4 | The duration of the tone in microseconds defined as 32-bit unsigned little-endian integer. |
| Volume | YES | | 1 | Playback volume as integer. |
| NumberOfRepeats | YES | | 1 | How many times to repeat the playback of the tone. Defined as integer. Zero value means that playback is done once, one means played two times, etc. |
| SilenceBetweenRepeats | YES | | 4 | The pause to keep between repeats defined as microseconds. Encoded as unsigned little-endian 32bit integer. |

**Response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x03 | 1 | PlayTone |
| OkResponse | YES | OK (8-bit | 2 | Response that the playback was completed successfully. |

| | | | | |
|---|---|---|---|---|
| | | string) | | |

### *4.4.8.4 Play DTMF*

Plays DTMF (Dual-Tone-Multi-Frequency) sounds as specified by the given string.

Response message is sent when the playback ends.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x04 | 1 | PlayDTMF |
| StringLength | YES | | 1 | Length of the DTMFString parameter in characters. |
| DTMFString | YES | | variable | DTMF string to be played. Encoded as 8-bit text. |
| ToneLength | YES | | 4 | Duration of one DTMF tone in microseconds. Encoded as unsigned little-endian 32bit integer. |
| GapLength | YES | | 4 | Gap between DTMF tones. Encoded as unsigned little-endian 32bit integer. |
| Volume | YES | | 1 | Playback volume as integer. |
| NumberOfRepeats | YES | | 1 | How many times to repeat the playback of the DTMF string. Defined as integer. Zero value means that playback is done once, one means played two times, etc. |
| SilenceBetweenRepeats | YES | | 4 | The pause to keep between repeats defined as microseconds. Encoded as unsigned little-endian 32bit integer. |

**Response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x04 | 1 | PlayDTMF |
| OkResponse | YES | OK (8-bit string) | 2 | Response that the playback was completed successfully. |

### *4.4.8.5 Stop*

Stops any currently playing file, tone or DTMF sequence.

When playback is stopped with the Stop command the OkResponse of the Play command is not sent. Only the OkResponse to the Stop command is sent.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x05 | 1 | Stop |

**Response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x05 | 1 | Stop |
| OkResponse | YES | OK (8-bit string) | 2 | Response that the playback was stopped. |

### *4.4.8.6 Get Duration*

Returns the duration of currently playing sound or the duration of the file specified as a parameter.

The command without parameters can be used only when a sound started with PlayFile command is playing. The command version with the file parameter can be used only when nothing is currently playing.

Getting the duration is not supported for rng ringtones, tones or DTMF strings.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x06 | 1 | GetDuration |
| PathLength | NO | | 1 | Length of the FullFilePath parameter in characters |
| FullFilePath | NO | | variable | Full path to the audio file whose duration to query. Encoded as 8-bit text. |

**Response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x06 | 1 | GetDuration |
| Duration | YES | | 4 | Duration of the audio file in microseconds. Encoded as unsigned little-endian 32bit integer. |

### *4.4.8.7 Get Max Volume*

Returns the maximum volume value that can be set for the currently playing sound of the file specified as a parameter.

The command without parameters can be used only when a sound is playing. The command version with the file parameter can be used only when nothing is currently playing.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x07 | 1 | GetMaxVolume |
| PathLength | NO | | 1 | Length of the FullFilePath parameter in characters. |
| FullFilePath | NO | | variable | Full path to the audio file whose maximum possible volume value to query. Encoded as 8-bit text. |

**Response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x07 | 1 | GetMaxVol |
| MaxVolume | YES | | 1 | The maximum volume value as integer. |

#### *4.4.8.8 Set Volume*

Command to change the current playback volume, it is effective only during playback.

If the specified value is greater than the maximum possible volume value then the volume is set to the maximum value.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x08 | 1 | SetVolume |
| Volume | YES | | 1 | Volume value to set as integer. |

**Response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x08 | 1 | SetVolume |
| Volume | YES | | 1 | The volume value that was set as integer. |

#### *4.4.8.9 Error response format*

In all error situations a framework error message (see 3.3) is returned instead of the normal responses that were defined in the above Sections.

**ServiceErrorDescription field contents**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | | 1 | Value is the code of the command that was being processed when this error |

| | | | | |
|---|---|---|---|---|
| | | | | occurred. |
| ErrorDescription | NO | 8-bit string | variable | Optional textual description of the error. |

The other fields of the error message are described in table 2 of the chapter 3.3.

### *4.4.9 PIM SERVICE*

The Personal Information Manager service enables importing contacts (vCards) and calendar events and to-do's (vCalendars) to phone.

The service supports the following commands:

| **Command/ Message Name** | **Command / Message Code** |
|---|---|
| Import vCard | 0x01 |
| Import vCalendar | 0x02 |
| Delete contact entries | 0x03 |
| Delete calendar entries | 0x04 |
| Add notepad memo | 0x05 |
| Add notepad memo from file | 0x06 |
| Delete all notepad memos | 0x07 |
| SIM card information | 0x10 |
| Import SIM contact | 0x11 |
| Delete SIM contact | 0x12 |
| Create bookmark | 0x1A |
| Delete bookmark | 0x1B |

The format for each command and response messages are defined in the following sub-chapters.

In case there is an error when executing the command a framework error message (see 3.3) is returned. In the error message the ServiceErrorCode contains a standard error code and the ServiceErrorDescription may contain textual description of the error.

#### *4.4.9.1 Import vCard*

This command imports one vCard, i.e. adds new contact entry to phone's contacts. If there is need to import multiple vCards, multiple service requests needs to be done.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x01 | 1 | Import vCard |
| vCard | YES | | variable | vCard in textual format |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |
| Entry id | YES | | 4 | Id of entry added to contacts. This is 32 bit signed integer. |

### *4.4.9.2 Import vCalendar*

This command imports one vCalendar, i.e. adds new event or to-do entry to phone's calendar. If there is need to import multiple vCalendars, multiple service requests are needed.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x02 | 1 | Import vCalendar |
| vCard | YES | | variable | vCalendar in textual format |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |
| Entry id | YES | | 4 | Id of entry added to calendar. This is 32 bit unsigned integer. |

### *4.4.9.3 Delete contact entries*

This command deletes one or all entries from the default contacts database. If the Entry id parameter is given, only that entry is deleted. If the Entry id parameter is omitted, all contact entries are deleted.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x03 | 1 | Delete contact entry/entries |
| Entry id | NO | | 4 | Id of the entry to be deleted from contacts. This is 32 bit signed integer. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.9.4 Delete calendar entries*

This command deletes one or all entries from the default calendar (agenda) database. If the Entry id parameter is given, only that entry is deleted. If the Entryid parameter is omitted, all calendar entries are deleted.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x04 | 1 | Delete calendar entry/entries |
| Entry id | NO | | 4 | Id of the entry to be deleted from calendar. This is 32 bit unsigned integer. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.9.5 Add notepad memo*

This command adds a memo to notepad.

Note that there are limits to the length of the message. Default value is about 10kb but this can be changed from HTI configuration files.

If large memos need to be inserted it is preferable to use the ‘add notepad from file’ command.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x05 | 1 | Add notepad memo |
| Text | YES | | variable | Contents of the memo as UTF-8 encoded string. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0x05 | 1 | Add notepad memo ok. |

### *4.4.9.6 Add notepad memo from file*

The command adds a memo to notepad from a file.

Note that there is a maximum size constraint to the files which can be added. More importantly if the file size is bigger than the allowed there will still be an ok response as the notepad API does not return an error to this.

Note that while the file path in the command uses UTF-8 encoding, the file contents should follow the UCS-2 encoding.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x06 | 1 | Add notepad memo from file |
| File path | YES | | variable | Path to a file in the device to add as a memo. UTF-8 encoded string. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0x06 | 1 | Add notepad memo from file ok. |

### *4.4.9.7 Delete all notepad memos*

This command will delete the whole notepad database file from the device thus removing all memos from notepad.

Notepad application needs to be closed for this command to succeed.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x07 | 1 | Delete all nopad memos |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0x07 | 1 | Delete notepad memos ok. |

### *4.4.9.8 SIM card information*

This command returns information about the currently inserted SIM card capabilities. This information can be used to find out what kind of contact fields are supported and valid for the import SIM contact command.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x10 | 1 | Get SIM card information |

**Ok response**

| Field | Mandatory | Value | Size (bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |
| Max num of second names | YES | 0x00 – 0xFF | 1 | The maximum amount of second name fields for one contact entry. |
| Max num of additional numbers | YES | 0x00 – 0xFF | 1 | The maximum amount of additional number fields for one contact entry. |
| Max num of e-mails | YES | 0x00 – 0xFF | 1 | The maximum amount of e-mail fields for one contact entry. |
| Max length of name | YES | 0x00 – 0xFF | 1 | The maximum length of a name field. |
| Max length of number | YES | 0x00 – 0xFF | 1 | The maximum length of a number field. |
| Max length of second name | YES | 0x00 – 0xFF | 1 | The maximum length of a second name field. |
| Max length of additional number | YES | 0x00 – 0xFF | 1 | The maximum length of an additional number field. |
| Max length of e-mail | YES | 0x00 – 0xFF | 1 | The maximum length of an e-mail field. |
| Total slots | YES | 0x0000 – 0xFFFF | 2 | The total amount of slots for contact entries in the SIM card (max number of contact entries). Defined as 16-bit little-endian integer. |
| Used slots | YES | 0x0000 – 0xFFFF | 2 | The amount of currently used contact entry slots. Defined as 16-bit little-endian integer. |

#### *4.4.9.9 Import SIM contact*

This command imports one contact entry to the SIM contacts directory. If there is need to import multiple contacts, multiple service requests needs to be done.

The supported field types, amount of fields and the maximum lengths of field data for different fields depend on the SIM card capabilities. There can always be one name field and one phone number field. Other capabilities can be queried by using the SIM card information command.

The “Type of field” parameter can have the following values:

| Field type | Value |
|---|---|
| Name field | 0x01 |
| Second name field | 0x02 |
| Phone number field | 0x03 |
| E-mail field | 0x04 |
| Additional number field | 0x05 |

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x11 | 1 | Add entry to SIM card contacts. |
| Amount of fields | YES | 0x00 – 0xFF | 1 | The amount of contact fields included in this entry. |
| Type of field | YES | 0x01 – 0x05 | 1 | The field type of the first field. |
| Field data length | YES | 0x00 – 0xFF | 1 | The length of the following data field. |
| Field data | YES | | 1 – 255 | The actual character data for this field. |
| …<br>Repeating the Type of field, Field data length and Field data parameters <Amount of fields> times. | | | | |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |
| Entry ID | YES | | 4 | ID of entry added to SIM contacts. Defined as a 32 bit little-endian integer. This ID can be used to delete the entry. |

### *4.4.9.10 Delete SIM contact*

This command deletes one or all entries from the SIM contacts directory. If the Entry id parameter is given, only that entry is deleted. If the Entryid parameter is omitted, all contact entries are deleted.

Note that if an error occurs during the delete all operation, an error message is sent but deleting some of the entries may still have succeeded.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x12 | 1 | Delete SIM contact entry/entries |
| Entry iD | NO | | 4 | ID of the entry to be deleted from SIM contacts. Defined as 32-bit little-endian integer. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.9.11 Create bookmark*

This command creates a browser bookmark and also creates a bookmark folder if the folder name is given and it doesn't exist.

If a bookmark with same name already exists in the folder an error message with ServiceErrorCode -11 (KErrAlreadyExists) is returned.

If Access Point name is given the access point must be found from the device, otherwise an error message with ServiceErrorCode -1 (KErrNotFound) is returned.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x1A | 1 | Create bookmark command. |
| Folder name length | YES | 0x00 – 0x32 | 1 | The length of the following folder name field. |
| Folder name | NO | | 0x00 – 0x32 | The name of the bookmark folder where the bookmark will be created. If the folder does not exist it will be created. If folder name is not given, bookmark is created to the root folder. |
| Bookmark name length | YES | 0x01 – 0x32 | 1 | The length of the following bookmark name field. |
| Bookmark name | YES | | 0x01 – 0x32 | The name of the bookmark to create. |
| URL length | YES | 0x0001-0x0400 | 2 | The length of the following URL field. |
| URL | YES | | 0x0001 – 0x0400 | The bookmark URL. |
| Access Point name length | YES | 0x00 – 0x1E | 1 | The length of the following access point name field. |
| Access Point name | NO | | 0x00 – 0x1E | The name of the access point of the bookmark. If not given, uses default. |
| User name length | YES | 0x00 – 0x28 | 1 | The length of the following user name field. |
| User name | NO | | 0x00 – 0x28 | The user name for the bookmark. |
| Password length | YES | 0x00 – 0x28 | 1 | The length of the following password field. |
| Password | NO | | 0x00 – | The password for the bookmark. |

| | | | | |
|---|---|---|---|---|
| | | | 0x28 | |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |
| Count of created items | YES | 0x01, 0x02 | 1 | How many items were created. 1 if only bookmark was created, 2 if also folder was created. |

### *4.4.9.12 Delete bookmark*

This command deletes a single browser bookmark or a bookmark folder (including all the bookmarks the folder contains).

If both the folder name and bookmark name are omitted, the command deletes all bookmark items and folders from the root (excluding the read only items).

If only folder name is specified the folder and all the bookmarks the folder contains will be deleted.

If only bookmark name is specified the bookmark is searched and deleted from the root folder.

If both folder and bookmark is specified the bookmark is searched and deleted from the named folder.

The second byte in the Ok response will always indicate the amount of items deleted.

Note that when deleting all items the delete count can also be zero if there were no items to delete. So this is not considered to be an error situation.

If non existing folder or bookmark name is specified the command will return an error message with ServiceErrorCode -1 (KErrNotFound).

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x1B | 1 | Delete bookmark command. |
| Folder name length | YES | 0x00 – 0x32 | 1 | The length of the following folder name field. |
| Folder name | NO | | 0x00 – 0x32 | The name of the bookmark folder to delete or the folder where the bookmark to be deleted is. |
| Bookmark name length | YES | 0x01 – 0x32 | 1 | The length of the following bookmark name field. |
| Bookmark name | NO | | 0x01 – 0x32 | The name of the bookmark to delete. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |
| Count of deleted items | YES | | 1 | How many items were deleted. Both a folder and a bookmark counts as one item. |

### *4.4.10 MESSAGES SERVICE*

Messages service enables the managing of the messages on the device.

The service supports the following commands:

| Command/ Message Name | Command / Message Code |
|---|---|
| Add SMS | 0x01 |
| Add MMS | 0x02 |
| Add E-Mail | 0x03 |
| Add IR Message | 0x04 |
| Add BT Message | 0x05 |
| Add Smart Message | 0x06 |
| Add Audio Message | 0x07 |
| Delete message | 0x10 |
| Delete folder content | 0x11 |
| Create Mailbox | 0x20 |
| Delete Mailbox | 0x21 |
| Create Access Point | 0x30 |
| Delete Access Point | 0x31 |
| Create Destination | 0x32 |
| Delete Destination | 0x33 |
| Add to Destination | 0x34 |
| Remove from Destination | 0x35 |
| Set default connection | 0x36 |
| Set Default SMS Center | 0x40 |
| Delete SMS Center | 0x41 |
| Set SMS Settings | 0x42 |

| | |
|---|---|
| Set MMS Settings | 0x45 |

The format for each command and response messages are defined in the following sub-chapters.

In case there is an error when executing the commands a framework error message (see 3.3) is returned.

#### *4.4.10.1 Add SMS*

This command creates a new SMS message to the given folder. The response contains a unique id for the created message that can be used later in deleting the message.

Note: If SMS is created to the outbox folder, HTI tries to send it immediately. For this the SMS Centre number must have been defined.

Service message consists of the following fields.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x01 | 1 | Add SMS message. |
| From/To field length | YES | >= 0 | 1 | Length of the following SMS message From/To field data. |
| From/To field value | NO | | variable | SMS message From/To field data value. Value is utf-8 formatted string. |
| Description field length | YES | >= 0 | 1 | Length of the following SMS message description data. |
| Description field value | NO | | variable | SMS message description data value. Value is utf-8 formatted string. |
| Message body length | YES | >= 0 | 2 | Length of the following SMS message body data. |
| Message body value | NO | | variable | SMS message body data value. Value is utf-8 formatted string. |
| Is new flag | YES | 0 or 1 | 1 | Flag indicating if the SMS message will be set new. |
| Is unread flag | YES | 0 or 1 | 1 | Flag indicating if the SMS message will be marked as unread. |
| Folder | YES | 0x01, 0x02, 0x03 or 0x04 | 1 | The destination folder for the new SMS message is specified according to the following enumeration:<br>0x01 = inbox,<br>0x02 = drafts,<br>0x03 = sent,<br>0x04 = outbox. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |
| Message id | YES | | 4 | Id of the created SMS message as 32-bit unsigned little-endian integer. |

#### *4.4.10.2 Add MMS*

This command creates a new MMS message to the given folder. The response contains a unique id for the created message that can be used later in deleting the message.

Note: If MMS message is created to the outbox folder, HTI tries to send it immediately. For this the MMS settings must have been defined.

Service message consists of the following fields.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x02 | 1 | Add MMS message. |
| From/To field length | YES | >= 0 | 1 | Length of the following MMS message From/To field data. |
| From/To field value | NO | | variable | MMS message From/To field data value. Value is utf-8 formatted string. |
| Description field length | YES | >= 0 | 1 | Length of the following MMS message description data. |
| Description field value | NO | | variable | MMS message description data value. Value is utf-8 formatted string. |
| Message body length | YES | >= 0 | 2 | Length of the following MMS message body data. |
| Message body value | NO | | variable | MMS message body data value. Value is utf-8 formatted string. |
| Attachment path length | YES | >=0 | 1 | Length of the following MMS message attachment path data. |
| Attachment path value | NO | | variable | MMS message attachment path data value. Value is utf-8 formatted string. |
| Is new flag | YES | 0 or 1 | 1 | Flag indicating if the MMS message will be set new. |
| Is unread flag | YES | 0 or 1 | 1 | Flag indicating if the MMS message will be marked as unread. |
| Folder | YES | 0x01, 0x02, 0x03 or 0x04 | 1 | The destination folder for the new MMS message is specified according to the following enumeration:<br>0x01 = inbox,<br>0x02 = drafts,<br>0x03 = sent,<br>0x04 = outbox. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |
| Message id | YES | | 4 | Id of the created MMS message as 32-bit unsigned little-endian integer. |

### *4.4.10.3 Add E-Mail*

This command creates a new E-Mail to the given folder. The response contains a unique id for the created E-Mail that can be used later in deleting the E-Mail. E-Mails created to Outbox folder are not sent automatically. They will be sent on next connection to the mailbox, or by selecting “Start” from Options menu in Outbox.

Note: If E-Mail is created to the outbox folder, HTI marks it to be sent immediately on next connection.

Note: The mailbox must exist before using this command.

Service message consists of the following fields.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x03 | 1 | Add E-Mail. |
| From/To field length | YES | >= 0 | 1 | Length of the following E-Mail From/To field data. |
| From/To field value | NO | | variable | E-Mail From/To field data value. Value is utf-8 formatted string. |
| Description field length | YES | >= 0 | 1 | Length of the following E-Mail description data. |
| Description field value | NO | | variable | E-Mail description data value. Value is utf-8 formatted string. |
| Message body length | YES | >= 0 | 2 | Length of the following E-Mail body data. |
| Message body value | NO | | variable | E-Mail body data value. Value is utf-8 formatted string. |
| Attachment path length | YES | >=0 | 1 | Length of the following E-Mail attachment path data. |
| Attachment path value | NO | | variable | E-Mail attachment path data value. Value is utf-8 formatted string. |
| Is new flag | YES | 0 or 1 | 1 | Flag indicating if the E-Mail will be set new. |
| Is unread flag | YES | 0 or 1 | 1 | Flag indicating if the E-Mail will be marked as unread. |
| Folder | YES | 0x01, 0x02, 0x03 or | 1 | The destination folder for the new E-Mail is specified according to the following enumeration: |

| | | | | |
|---|---|---|---|---|
| | | 0x04 | | 0x01 = inbox,<br>0x02 = drafts,<br>0x03 = sent,<br>0x04 = outbox. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |
| Message id | YES | | 4 | Id of the created E-Mail as 32-bit unsigned little-endian integer. |

### *4.4.10.4 Add IR Message*

This command creates a new IR message to the given folder. The response contains a unique id for the created message that can be used later in deleting the message.

Note: Outbox folder is not supported for IR messages.

Service message consists of the following fields.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x04 | 1 | Add IR message. |
| From/To field length | YES | >= 0 | 1 | Length of the following IR message From/To field data. |
| From/To field value | NO | | variable | IR message From/To field data value. Value is utf-8 formatted string. |
| Description field length | YES | >= 0 | 1 | Length of the following IR message description data. |
| Description field value | NO | | variable | IR message description data value. Value is utf-8 formatted string. |
| Attachment path length | YES | >=0 | 1 | Length of the following IR message attachment path data. |
| Attachment path value | NO | | variable | IR message ttachment path data value. Value is utf-8 formatted string. |
| Is new flag | YES | 0 or 1 | 1 | Flag indicating if the IR message will be set new. |
| Is unread flag | YES | 0 or 1 | 1 | Flag indicating if the IR message will be marked as unread. |
| Folder | YES | 0x01, 0x02, 0x03 or 0x04 | 1 | The destination folder for the new IR message is specified according to the following enumeration:<br>0x01 = inbox,<br>0x02 = drafts,<br>0x03 = sent,<br>0x04 = outbox. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |
| Message id | YES | | 4 | Id of the created IR message as 32-bit unsigned little-endian integer. |

### *4.4.10.5 Add BT Message*

This command creates a new BT message to the given folder. The response contains a unique id for the created message that can be used later in deleting the message.

Note: Outbox folder is not supported for BT messages.

Service message consists of the following fields.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x05 | 1 | Add BT message. |
| From/To field length | YES | >= 0 | 1 | Length of the following BT message From/To field data. |
| From/To field value | NO | | variable | BT message From/To field data value. Value is utf-8 formatted string. |
| Description field length | YES | >= 0 | 1 | Length of the following BT message description data. |
| Description field value | NO | | variable | BT message description data value. Value is utf-8 formatted string. |
| Attachment path length | YES | >=0 | 1 | Length of the following BT message attachment path data. |
| Attachment path value | NO | | variable | BT message attachment path data value. Value is utf-8 formatted string. |
| Is new flag | YES | 0 or 1 | 1 | Flag indicating if the BT message will be set new. |
| Is unread flag | YES | 0 or 1 | 1 | Flag indicating if the BT message will be marked as unread. |
| Folder | YES | 0x01, 0x02, 0x03 or 0x04 | 1 | The destination folder for the new BT message is specified according to the following enumeration:<br>0x01 = inbox,<br>0x02 = drafts,<br>0x03 = sent,<br>0x04 = outbox. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |
| Message id | YES | | 4 | Id of the created BT message as 32-bit unsigned little-endian integer. |

### *4.4.10.6 Add Smart Message*

This command creates a new smart message to inbox folder. No other folders are allowed with this command. The response contains a unique id for the created message that can be used later in deleting the message.

Service message consists of the following fields.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x06 | 1 | Add smart message. |
| From/To field length | YES | >= 0 | 1 | Length of the following smart message From/To field data. |
| From/To field value | NO | | variable | Smart message From/To field data value. Value is utf-8 formatted string. |
| Description field length | YES | >= 0 | 1 | Length of the following smart message description data. |
| Description field value | NO | | variable | Smart message description data value. Value is utf-8 formatted string. |
| Message body length | YES | >= 0 | 2 | Length of the following smart message body data. |
| Message body value | NO | | variable | Smart message body data value. Value is utf-8 formatted string. |
| Is new flag | YES | 0 or 1 | 1 | Flag indicating if the smart message will be set new. |
| Is unread flag | YES | 0 or 1 | 1 | Flag indicating if the smart message will be marked as unread. |
| Folder | YES | 0x01, 0x02, 0x03 or 0x04 | 1 | The destination folder for the new SMS message is specified according to the following enumeration:<br>0x01 = inbox,<br>0x02 = drafts,<br>0x03 = sent,<br>0x04 = outbox. |
| BIO message type UID | YES | | 4 | The UID defining the type of the created BIO message as 32-bit unsigned little-endian integer according to the following enumeration:<br>0x1000552F = Internet AP Settings,<br>0x10005530 = E-Mail Notification,<br>0x10005531 = Business Card,<br>0x10005532 = WAP AP Settings,<br>0x10005533 = VCalendar Entry, |

| | | | | |
|---|---|---|---|---|
| | | | | 0x10005534 = VCard Entry,<br>0x10005535 = Ringing Tone,<br>0x10005536 = Operator Logo,<br>0x1000125D = WAP Provisioning,<br>0x10005269 = CLI Logo. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |
| Message id | YES | | 4 | Id of the created smart message as 32-bit unsigned little-endian integer. |

### *4.4.10.7 Add Audio Message*

This command creates a new audio message (a special type of MMS) to the given folder. The response contains a unique id for the created message that can be used later in deleting the message.

Compared to the normal MMS message, the audio message does not have any body text element and the attachment (the AMR audio file) is mandatory.

Note: If audio message is created to the outbox folder, HTI tries to send it immediately. For this the MMS settings must have been defined.

Service message consists of the following fields.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x07 | 1 | Add Audio message. |
| From/To field length | YES | >= 0 | 1 | Length of the following audio message From/To field data. |
| From/To field value | NO | | variable | Audio message From/To field data value. Value is utf-8 formatted string. |
| Description field length | YES | >= 0 | 1 | Length of the following audio message description data. |
| Description field value | NO | | variable | Audio message description data value. Value is utf-8 formatted string. |
| Attachment path length | YES | >=4 | 1 | Length of the following audio message attachment path data. |
| Attachment path value | YES | | variable | Audio message attachment path data value. Value is utf-8 formatted string. |
| Is new flag | YES | 0 or 1 | 1 | Flag indicating if the audio message will be set new. |
| Is unread flag | YES | 0 or 1 | 1 | Flag indicating if the audio message |

| | | | | |
|---|---|---|---|---|
| | | | | will be marked as unread. |
| Folder | YES | 0x01, 0x02, 0x03 or 0x04 | 1 | The destination folder for the new audio message is specified according to the following enumeration:<br>0x01 = inbox,<br>0x02 = drafts,<br>0x03 = sent,<br>0x04 = outbox. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |
| Message id | YES | | 4 | Id of the created audio message as 32-bit unsigned little-endian integer. |

### *4.4.10.8 Delete Message*

This command deletes a single message according to the given message id.

Service message consists of the following fields.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x10 | 1 | Delete message |
| Message id | YES | | 4 | The id identifying the message to be deleted defined as 32-bit unsigned little-endian integer. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.10.9 Delete Folder Content*

This command deletes all messages of specified type from the specified folder.

Service message consists of the following fields.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x11 | 1 | Delete folder content |

| | | | | |
|---|---|---|---|---|
| Folder | YES | 0x00, 0x01, 0x02, 0x03 or 0x04 | 1 | The folder whose content is to be deleted is specified according to the following enumeration:<br>0x00 = all,<br>0x01 = inbox,<br>0x02 = drafts,<br>0x03 = sent,<br>0x04 = outbox. |
| Message type | YES | 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06 or 0x07 | 1 | The type of messages to be deleted is specified according to the following enumeration:<br>0x00 = all,<br>0x01 = SMS,<br>0x02 = MMS,<br>0x03 = Smart Message,<br>0x04 = E-Mail,<br>0x05 = IR Message,<br>0x06 = BT Message,<br>0x07 = Audio Message |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.10.10 Create Access Point*

This command creates an internet access point.

The following table lists the codes for different bearer types. Note that the supported bearer types depend on the platform and device.

| Bearer type | Value |
|---|---|
| Data call (CSD) | 0x01 |
| Packet data (GPRS) | 0x02 |
| Data call (HSCSD) | 0x04 |
| Packet data (CDMA) | 0x10 |
| Wireless LAN | 0x20 |

The following table lists the field type codes for different access point setting fields. The field type code values are based on the TApMember enum defined in ApAccessPointItem.h header file.

Different fields are used with different bearer types. The possible fields for each bearer type can best be seen from the device's access point creation UI. Note that normally you can leave most of the fields to their default value, so you don't need to include them in the create command.

| Field type | Enum (TApMember) | Field | Possible values |
|---|---|---|---|

| | | type code | |
|---|---|---|---|
| Access point name | EApGprsAccessPointName | 58 | |
| User name | EApIspLoginName | 23 | |
| Prompt password | EApIspIfPromptForAuth | 29 | 0 = no, 1 = yes |
| Password | EApIspLoginPass | 24 | |
| Password authentication | EApIspDisablePlainTextAuth | 46 | 0 = normal, 1 = secure |
| Starting page | EApWapStartPage | 3 | |
| Phone IP address | EApIspIPAddr | 38 | |
| Primary Name Server | EApIspIPNameServer1 | 42 | |
| Secondary Name Server | EApIspIPNameServer2 | 43 | |
| Proxy server address | EApProxyServerAddress | 91 | |
| Port number | EApProxyPortNumber | 93 | |
| WLAN Network name | EApWlanNetworkName | 163 | |
| WLAN network mode | EApWlanNetworkMode | 164 | 0 = AdHoc<br>1 = Infrastructure |
| WLAN security mode | EApWlanSecurityMode | 165 | 1 = Open network<br>2 = WEP<br>4 = 802.1x<br>8 = WPA<br>16 = WPA2 |
| Primary Name Server | EApIP6NameServer1 | 104 | |
| Secondary Name Server | EApIP6NameServer2 | 105 | |
| Proxy server address | EApProxyServerAddress | 91 | |
| Port number | EApProxyPortNumber | 93 | |
| Access number | EApIspDefaultTelNumber | 18 | |
| Data call type | EApIspBearerCallTypeIsdn | 50 | 0 = Analogue<br>1 = ISDN v.110<br>2 = ISDN v.120 |
| Max connection speed | EApIspBearerSpeed | 49 | 0 = Auto detect<br>1 = 9600, 2 = 14400, 3 = 19200, 4 = 28800, 5 = 38400, 6 = 43200, 7 = 56000 |
| Use call back | EApIspIfCallbackEnabled | 33 | 0 = no, 1 = yes |
| Call back type | EApIspIfCallbackType | 34 | See Enum TCallbackAction from nifvar.h header file. |
| Call back number | EApIspIfCallbackInfo, | 35 | |
| Enable PPP compression | EApIspEnableIpHeaderComp | 44 | 0 = no, 1 = yes |
| Use login script | EApIspUseLoginScript | 20 | 0 = no, 1 = yes |
| Login script | EApIspLoginScript | 21 | |
| Modem init string | EApIspInitString | 52 | |

| | | | |
|---|---|---|---|
| Network type | EApGprsPdpType | 59 | 0 = IPv4, 1 = IPv6 |
| Use access point | Field does not exist in TApMember | 200 | 1 = After confirmation, 2 = Automatically |

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x30 | 1 | Create access point command. |
| Bearer type | YES | | 1 | Type of bearer for this access point. |
| Name field length | YES | > 0 | 1 | Length of the following name data. |
| Name field value | YES | | | The name of the access point to create. |
| Amount of fields | YES | | 1 | The amount of setting fields in this command. |
| Field type code | YES | | 1 | The field type code of the first field. |
| Field data length | YES | > 0 | 1 | The length of the following data field. |
| Field data | YES | | 1 – 255 | The actual data for this field as 8-bit text. Also numerical values must be as text. |
| …<br>Repeating the Field type, Field data length and Field data parameters <Amount of fields> times. | | | | |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |
| AP id | YES | | 4 | Id of the created access point as 32-bit unsigned little-endian integer. |

### *4.4.10.11 Delete Access Point*

This command deletes an internet access point based on access point name. If the access point being deleted is in use, the connections using the access point will be closed and then the access point is deleted.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x31 | 1 | Delete access point command. |
| Name field length | YES | >= 0 | 1 | Length of the following name data. |
| Name field value | YES | | | The name of the access point to |

| | | | | |
|---|---|---|---|---|
| | | | | delete. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.10.12 Create Destination*

This command creates a new Destination in connection settings.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x32 | 1 | Create destination command. |
| Name field length | YES | >= 0 | 1 | Length of the following name data. |
| Destination name | YES | | | The name of the Destination to create. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.10.13 Delete Destination*

This command deletes a Destination in connection settings

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x33 | 1 | Delete destination command. |
| Name field length | YES | >= 0 | 1 | Length of the following name data. |
| Destination name | YES | | | The name of the Destination to delete. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.10.14 Add to Destination*

This command adds (moves) the named internet access point (connection method) to a named Destination.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x34 | 1 | Add to Destination command. |
| Name field length | YES | >= 0 | 1 | Length of the following name data. |
| Access Point name | YES | | | The name of the Access Point to add. |
| Name field length | YES | >=0 | 1 | Length of the following name data. |
| Destination name | YES | | | The name of the Destination where to add the Access Point. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.10.15 Remove from Destination*

This command removes the named internet access point (connection method) from a named Destination to Uncategorised.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x35 | 1 | Remove from Destination command. |
| Name field length | YES | >= 0 | 1 | Length of the following name data. |
| Access Point name | YES | | | The name of the Access Point to remove. |
| Name field length | YES | >=0 | 1 | Length of the following name data. |
| Destination name | YES | | | The name of the Destination from where to remove the Access Point. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.10.16 Set default connection*

This command sets the default connection setting. Currently the setting can be “Always ask”, “Ask once” or a specified Destination or Access Point (Connection Method) can be set as default.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x36 | 1 | Set default connection command. |
| Default connection setting | YES | 0x00 – 0x03 | 1 | 0 = Always ask, 1 = Ask once, 2 = Named Destination, 3 = Named Access Point |
| Name field length | YES | | 1 | Length of the following name data. If the above setting is 0 or 1 then this shoul be zero and the following name field omitted. |
| Access Point / Destination name | NO | | | The name of the Destination or Access Point to set as default connection.<br>If the above Default connection setting is 2 then this must be a Destination name, if it’s 3 then an Access Point name. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.10.17 Create Mailbox*

This command creates a new remote mailbox configuration.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x20 | 1 | Create mailbox command. |
| Mailbox type | YES | 0x00, 0x01 | 1 | The type of mailbox to create: 0 = POP3 account 1 = IMAP4 account. |
| Mailbox name length | YES | 0x01 – 0x1E | 1 | The length of the following name field. |
| Mailbox name | YES | | 1 – 30 | The name of the mailbox to create. Must be unique, another mailbox with same name must not exist. |
| Incoming mail server name length | YES | 0x03 – 0x32 | 1 | The length of the following name field. |
| Incoming mail server name | YES | | 3 – 50 | The domain name of the incoming mail server. |
| Incoming access point | YES | 0x00 – | 1 | The length of the following name |

| | | | | |
|---|---|---|---|---|
| name length | | 0xFF | | field. |
| Incoming access point name | NO | | 0 – 255 | The name of the access point to use when connecting to incoming mail server. If this is omitted, the AP setting will be “Always ask”. |
| Incoming user name length | YES | 0x00 – 0x32 | 1 | The length of the following name field. |
| Incoming user name | NO | | 1 – 50 | The user name for login to incoming mail server. |
| Incoming password length | YES | 0x00 – 0x32 | 1 | The length of the following password field. |
| Incoming password | NO | | 1 – 50 | The password for login to incoming mail server. |
| Incoming security | YES | 0x00 – 0x02 | 1 | The security setting for incoming mail server: 0 = 0ff, 1 = TLS, 2 = SSL |
| Incoming port | YES | 0x0000 – 0x03E7 | 2 | The port number for incoming mail server ( 0 = use default ) |
| A POP secure login | YES | 0x00 – 0x01 | 1 | Use secure login for POP account. 0 = no, 1 = yes. Ignored if creating IMAP account. |
| Outgoing mail server name length | YES | 0x03 – 0x32 | 1 | The length of the following name field. |
| Outgoing mail server name | YES | | 3 – 50 | The domain name of the outgoing mail server. |
| Outgoing access point name length | YES | 0x00 – 0xFF | 1 | The length of the following name field. |
| Outgoing access point name | NO | | 0 – 255 | The name of the access point to use when connecting to outgoing mail server. If this is omitted, the AP setting will be “Always ask”. |
| Outgoing user name length | YES | 0x00 – 0x32 | 1 | The length of the following name field. |
| Outgoing user name | NO | | 1 – 50 | The user name for login to outgoing mail server. |
| Outgoing password length | YES | 0x00 – 0x32 | 1 | The length of the following password field. |
| Outgoing password | NO | | 1 – 50 | The password for login to outgoing mail server. |
| Outgoing security | YES | 0x00 – 0x02 | 1 | The security setting for outgoing mail server: 0 = 0ff, 1 = TLS, 2 = SSL |
| Outgoing port | YES | 0x0000 – 0x03E7 | 2 | The port number for outgoing mail server ( 0 = use default ) |
| Own mail address length | YES | 0x01 – 0x64 | 1 | The length of the following mail address field. |
| Own mail address | YES | | 1 – 100 | Own mail address. |

| | | | | |
|---|---|---|---|---|
| Own name length | YES | 0x00 – 0x64 | 1 | The length of the following name field. |
| Own name | NO | | 0 – 100 | Name to be shown as the sender of outgoing messages. |
| Send message option | YES | 0x00 – 0x02 | 1 | When to send outgoing messages: 0 = immediately, 1 = on next connection, 2 = on request |
| Copy to own address | YES | 0x00 – 0x03 | 1 | Whether to send a copy of outgoing messages to own address: 0 = no copy, 1 = in to field, 2 = in cc field, 3 = in bcc field |
| Sinature text length | YES | 0x0000 – 0x01F4 | 2 | The length of the following signature text field. |
| Signature text | NO | | 0 – 500 | If defined the signature text will be added to outgoing messages. |
| New mail indicators | YES | 0x00 – 0x01 | 1 | Show new mail indicators: 0 = no, 1 = yes |
| Retrieved parts | YES | 0x00 – 0x02 | 1 | Which parts of incoming messages to retrieve: 0 = only headers, 1 = less than defined size, 2 = body and attachments. Ignored if creating IMAP account. |
| Retrieve size limit | YES | 0x0000 – 0x03E7 | 2 | The maximum size of incoming messages to retrieve (kilo bytes). Ignored if Retrieved parts option is not 1. |
| Emails to retrieve | YES | 0x0000 – 0x03E7 | 2 | Maximum number of incoming messages to retrieve. |
| IMAP4 folder path length | YES | 0x00 – 0x64 | 1 | The length of the following path field. |
| IMAP4 folder path | NO | | 0 – 100 | The IMAP folder path. Ignored if creating POP account. |
| Automatic update | YES | 0x00 – 0x02 | 1 | Automatic update setting: 0 = always on, 1 = only in home network, 2 = not enabled |
| Set as default | YES | 0x00 – 0x01 | 1 | Set this account as default for outgoing messages. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.10.18 Delete Mailbox*

This command deletes a remote mailbox configuration based on the mailbox name.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x21 | 1 | Delete mailbox command. |
| Name field length | YES | >= 0 | 1 | Length of the following name data. |
| Name field value | YES | | | The name of the mailbox to delete. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.10.19 Set Default SMS Centre*

This command sets the default SMS centre using the following logic:

- If there already is a SMS centre with the same name and number, it is set to default SMS centre
- If there already is a SMS centre with the same name and different number, an error is returned (no changes to existing SMS centres are made)
- If no SMS centre with the given name is found, a new SMS centre is created and it is set to default SMS centre

Note: SMS Settings view should be closed when this command is executed. If it is open this command may not have any effect even if it returns an OK response.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x40 | 1 | Set default SMS Center command. |
| SMS center name field length | YES | >= 0 | 1 | Length of the following SMS center name data. |
| SMS center name field value | YES | | | The name of the SMS center. |
| SMS center number field length | YES | >=0 | 1 | Length of the following SMS center number data. |
| SMS center number field value | YES | | | The number of the SMS center. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.10.20 Delete SMS Centre*

This command deletes the SMS centre with a given name. If no SMS centre with the given name is found, an error is returned.

Note: SMS Settings view should be closed when this command is executed. If it is open this command may not have any effect even if it returns an OK response.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x41 | 1 | Delete SMS Center command. |
| SMS center name field length | YES | >= 0 | 1 | Length of the following SMS center name data. |
| SMS center name field value | YES | | | The name of the SMS center. |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.10.21 Set SMS Settings*

This command sets the values for SMS settings.

Note: SMS Settings view should be closed when this command is executed. If it is open this command may not have any effect even if it returns an OK response.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x42 | 1 | Set SMS Settings command. |
| Character support | YES | 0x00 – 0x01 | 1 | 0 = Full,<br>1 = Reduced |
| Delivery report | YES | 0x00 – 0x01 | 1 | 0 = No,<br>1 = Yes |
| Validity period | YES | 0x00 – 0x05 | 1 | 0 = Maximum,<br>1 = 1 hour,<br>2 = 6 hours,<br>3 = 24 hours,<br>4 = 3 days, |

| | | | | |
|---|---|---|---|---|
| | | | | 5 = 1 week |
| Message conversion | YES | 0x00 – 0x03 | 1 | 0 = None,<br>1 = Fax,<br>2 = Paging,<br>3 = Email |
| Preferred connection | YES | 0x00 – 0x01 | 1 | 0 = GSM,<br>1 = Packet data |
| Reply via same center | YES | 0x00 – 0x01 | 1 | 0 = No,<br>1 = Yes |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

### *4.4.10.22 Set MMS Settings*

This command sets the values for MMS service settings.

Note: The messages application should be closed when this command is executed. If it is open then this settings change command may not have any effect even if it returns an OK response.

**Syntax**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| CommandCode | YES | 0x45 | 1 | Set MMS settings command. |
| Access point name field length | YES | 0x01 – 0x1E | 1 | Length of the following access point name data. |
| Access point name field value | YES | | 1 - 30 | The name of the access point to be used for MMS sending and receiving. |
| MMS creation mode | YES | 0x00 – 0x02 | 1 | 0 = restricted, 1 = guided, 2 = free |
| Image size | YES | 0x00 – 0x02 | 1 | 0 = small, 1 = large, 2 = original<br>Original is not allowed if MMS creation mode is set to restricted. |
| MMS reception | YES | 0x00 – 0x03 | 1 | 0 = always automatic, 1 = automatic in home network, 2 = always manual, 3 = disabled |
| Receive anonymous messages | YES | 0x00 – 0x01 | 1 | 0 = no, 1 = yes |
| Receive advertisements | YES | 0x00 – 0x01 | 1 | 0 = no, 1 = yes |
| Request delivery reports | YES | 0x00 – 0x01 | 1 | 0 = no, 1 = yes |
| Send delivery reports | YES | 0x00 – 0x01 | 1 | 0 = no, 1 = yes |

| | | | | |
|---|---|---|---|---|
| Validity period | YES | 0x00 – 0x05 | 1 | 0 = maximum, 1 = 1 hour, 2 = 6 hours, 3 = 24 hours, 4 = 3 days, 5 = 1 week |

**Ok response**

| Field | Mandatory | Value | Size(bytes) | Description |
|---|---|---|---|---|
| Ok status | YES | 0xFF | 1 | EResultOk = 0xFF |

# 5 FUTURE WORK

This chapter captures all those items to be captured in follow-up work including further work to be done by other standards organisations as applicable.

- Navigation Aid
- Enhanced HTI command-set

# 6 DEFINITION OF TERMS

| TERM | DESCRIPTION |
|---|---|
| **FUNCTION** | Term used within context of test service. Each test service may have several functionalities. For example, test service named “HTI” has conceptually many system functions like authentication and device boot. |
| **HARD RESET** | Resetting the Terminal to the state it would be in during normal use following the removal of power (disconnection of battery or external supply), the reconnection of power and pressing the on button (long press if necessary). |
| **LOCAL ACCESS** | Test access to the Terminal over a short distance without using long-range transportation means, e.g. GSM, WLAN. |
| **MASTER RESET** | Resetting the Terminal to a factory default state. All additionally installed content (e.g. applications, images) are erased and connection profiles are set to factory default. |
| **SOFT RESET** | Resetting the Terminal to the state it would be in normal use following the pressing of an “off” button and then pressing the “on” button. |
| **TERMINAL** | Used as an alternative term for a cellular telephone, device, mobile phone, phone or handset. |
| **TEST CONTROL** | The test control is a (wired) connection to a local PC which controls the device via AT commands or HTI. |
| **WIRED LINK** | A connection between the Terminal and another device (e.g. PC) using a cable. |
| **PRE-INSTALLED** | As Pre-Installed application are considered applications which are natively part of the OS or directly deployed after the first boot of the Terminal (including first boot up after a Master Reset). |

# 7 ABBREVIATIONS

| ABBREVIATION | DESCRIPTION |
|---|---|
| **AEE** | Application Execution Environment |
| **AMR** | Adaptive Multi Rate |
| **API** | Application Programming Interface |
| **ASCII** | American Standard Code for Information Exchange |
| **ASF** | Application Security Framework |
| **AT** | Attention |
| **BMP** | Basic Multilingual Plane |
| **BT** | Bluetooth |
| **CRC** | Cyclic Redundancy Check |
| **DLL** | Dynamic Linked Library |
| **DTMF** | Dual Tone Multi-Frequency |
| **ESN** | Equipment Serial Number |
| **FTP** | File Transfer Protocol |
| **GCF** | Global Certification Forum |
| **GSM** | Global System for Mobile communications |
| **HTI** | Harmonised Test Interface |
| **IMEI** | International Mobile Equipment Identifier |
| **Ir** | Infra-Red |
| **LIBCAP** | Library of Capabilities |
| **MIDP** | Mobile Information Device Profile |
| **MIME** | Multipurpose Internet Mail Extensions |

| ABBREVIATION | DESCRIPTION |
|---|---|
| **MMS** | Multimedia Message Service |
| **MTR** | Global System for Mobile |
| **MTT** | Mobile Terminal Testing |
| **OMSI** | Open Mobile Service Interface |
| **PIM** | Personal Information Manager |
| **RSSI** | Received Strength Signal Indicator |
| **SMS** | Short Message Service |
| **TE** | Terminal Equipment (equal to DCE – Data Circuit terminating Equipment), e.g. GSM data card – [3] Chapter 3.2 |
| **TA** | Terminal Adaptor (equal to DTE – Data Terminal Equipment), e.g. a computer – [3] Chapter 3.2 |
| **UCS-2** | 2 – byte Universal Character Code |
| **UI** | User Interface |
| **UID** | Unique Identifier |
| **UMTS** | Universal Mobile Telecommunications System |
| **USB** | Universal Serial Bus |
| **UTF-8** | 8-bit UCS/Unicode Transfer Format |
| **WLAN** | Wireless Local Area Network |
| **XML** | EXtensible Markup Language |

# 8 REFERENCED DOCUMENTS

| NO. | DOCUMENT | AUTHOR | DATE |
|---|---|---|---|
| **1** | RFC 2119 - Key words for use in RFCs to Indicate Requirement Levels | IETF | |
| **2** | OMTP Application Security Framework, v2.2 | OMTP | 2008-06 |
| **3** | 3GPP TS27.007 V8.6.<br><br>AT Command set for User Equipment (UE),<br><br>Release 8 | 3GPP | 2008-06 |
| **4** | 3GPP TS27.005 V8.0.0<br><br>Use of Data Terminal Equipment – Data Circuit terminating Equipment (DTE – DCE) interface for Short Message Service (SMS) and Cell Broadcast Service (CBS)<br><br>Release 8 | 3GPP | 2008-03 |
| **5** | ITU-T V.250<br><br>Series V: Data communication over the telephone network<br><br>Serial asynchronous automatic dialling and control | ITU-T | 2003-07 |

# 9 APPENDICES

## 9.1 NEW AT COMMANDS REQUIREMENTS

### 9.1.1 RESET CONTROL – CRST

| COMMAND | POSSIBLE RESPONSE(S) |
|---|---|
| **+CRST={soft , hard}** | +CME ERROR: <err> |
| **+CRST=?** | +CRST=list of supported reset types |

Description:

This command requests the Terminal to reset itself. The command supports two modes of reset: soft reset and hard reset.

Hard reset: A hard reset erases all information in memory.

Soft reset: Soft reset initialises various Terminal functions but does not reset the memory.

Implementations might support only hard reset type.

### 9.1.2 BATTERY CHARGE – CBC

| COMMAND | POSSIBLE RESPONSE(S) |
|---|---|
| **+CBC** | +CBC: <bcs>,<bcl><br>+CME ERROR: <err> |
| **+CBC=?** | +CBC: (list of supported <bcs>s), (list of supported <bcl>s) |
| **+CBC=<bcs>** | +CME Error: <err> |

Description:

The addition to this existing command is the ability to control the charging of the MT. Available bcs values are:

0 – Terminal is only powered by the battery (charging disabled)

1 – Terminal is powered by battery and charger (charging enabled)

The rest of the bcs values (2,3) are not relevant to the charging control.

### 9.1.3 WCDMA SIGNAL QUALITY +CWCDMASQ

| COMMAND | POSSIBLE RESPONSE(S) |
|---|---|
| **+CWCDMASQ** | +CWCDMASQ: <rssi>,<rscp>,<ecio><br><br>*+CME ERROR: <err>* |
| **+CWCDMASQ =?** | +CWCDMASQ: (list of supported <rssi>s),(list of supported <rscp>s),(list of supported <ecio>s) |

**Description**

Execution command returns UTRA Carrier RSSI <rssi>, CPICH received signal code power `<rscp>` and CPICH EcIo `<ecio>` from the Terminal.

**Defined values**

`<rssi>:`

Returns integer value as per 3GPP 23.133 Section 9.1.3.3

For the purpose of this command some additional values are added (in red below) for completeness.

| Reported value | Measured quantity value | Unit |
|---|---|---|
| UTRA_carrier_RSSI_LEV_00 | UTRA carrier RSSI < -100 | dBm |
| UTRA_carrier_RSSI_LEV_01 | -100 ≤ UTRA carrier RSSI < -99 | dBm |
| UTRA_carrier_RSSI_LEV_02 | -99 ≤ UTRA carrier RSSI < -98 | dBm |
| … | … | … |
| UTRA_carrier_RSSI_LEV_74 | -27 ≤ UTRA carrier RSSI < -26 | dBm |
| UTRA_carrier_RSSI_LEV_75 | -26 ≤ UTRA carrier RSSI < -25 | dBm |
| UTRA_carrier_RSSI_LEV_76 | -25 ≤ UTRA carrier RSSI | dBm |
| … | … | … |
| 77 to 98 | undefined | n/a |
| … | … | … |
| 99 | not known or not detectable | n/a |

<rscp>:

Returns integer value as per 3GPP 23.133 Section 9.1.1.3

For the purpose of this command some additional values are added (in red below) for completeness.

| Reported value | Measured quantity value | Unit |
|---|---|---|
| -05 | CPICH RSCP <-120 | dBm |
| -04 | -120 ≤ CPICH RSCP < -119 | dBm |
| -03 | -119 ≤ CPICH RSCP < -118 | dBm |
| … | … | … |
| 89 | -27 ≤ CPICH RSCP < -26 | dBm |
| 90 | -26 ≤ CPICH RSCP < -25 | dBm |
| 91 | -25 ≤ CPICH RSCP | dBm |
| …. | … | … |
| 92 to 98 | undefined | n/a |
| … | … | … |
| 99 | not known or not detectable | n/a |

<EcIo>:

Returns integer value as per 3GPP 23.133 Section 9.1.2.3

For the purpose of this command some additional values are added (in red below) for completeness.

| Reported value | Measured quantity value | Unit |
|---|---|---|
| 00 | CPICH Ec/Io < -24 | dB |
| 01 | -24 ≤ CPICH Ec/Io < -23.5 | dB |
| 02 | -23.5 ≤ CPICH Ec/Io < -23 | dB |
| … | … | … |
| 47 | -1 ≤ CPICH Ec/Io < -0.5 | dB |
| 48 | -0.5 ≤ CPICH Ec/Io < 0 | dB |
| 49 | 0 ≤ CPICH Ec/Io | dB |
| … | … | … |
| 50 to 98 | undefined | n/a |
| … | … | … |
| 99 | not known or not detectable | n/a |

### *9.1.4 UE TRANSMITTED POWER +CTXPWR*

| COMMAND | POSSIBLE RESPONSE(S) |
|---|---|
| **+CTXPWR** | + CTXPWR: <txpwr><br><br>*+CME ERROR: <err>* |
| **+CTXPWR =?** | +CTXPWR: (list of supported <txpwr>s) |

**Description**

Execution command  returns UE transmitted power.

**Defined values**

<txpwr>:

Returns integer value as per 3GPP 23.133 Section 9.1.6.2.

For the purpose of this command some additional values are added (in red below) for completeness.

| Reported value | Measured quantity value (dBm) |
|---|---|
| 104 | 33<= to <34 |
| 103 | 32<= to <33 |
| 102 | 31<= to <32 |
| … | … |
| 096 | 25<= to <26 |
| 095 | 24<= to <25 |
| 094 | 23<= to <24 |
| 093 | 22<= to <23 |
| 092 | 21<= to <22 |
| 091 | 20<= to < 21 |
| 090 | 19<= to <20 |
| 089 | 18<= to <19 |
| 088 | 17<= to <18 |
| 087 | 16<= to <17 |
| 086 | 15<= to <16 |
| 085 | 14<= to <15 |
| 084 | 13<= to <14 |
| 083 | 12<= to <13 |
| 082 | 11<= to <12 |
| 081 | 10<= to <11 |
| … | … |
| 023 | -48<= to <-47 |
| 022 | -49<= to <-48 |
| 021 | -50<= to <-49 |
| .... | … |
| 20 to 1 | undefined |
| … | … |
| 0 | not known or not detectable |

**-------- END OF DOCUMENT --------**