
**Information technology — Open Terminal
Architecture (OTA) — Virtual machine**

*Technologies de l'information — Architecture des terminaux ouverte
(OTA) — Machine virtuelle*

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2010

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword	vi
Introduction.....	vii
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Symbols and abbreviated terms	5
5 Data types, stack notation and flags	6
5.1 Data Types	6
5.2 Stack Notation	7
5.3 Flags	7
6 OTA virtual machine.....	7
6.1 General principles	7
6.2 Virtual Machine CPU	8
6.2.1 Registers	9
6.2.2 Virtual Machine Size and Cells.....	9
6.2.3 Memory	9
6.2.4 Stacks	10
6.2.5 Frame Mechanism and Usage.....	11
6.2.6 Extensible Memory.....	12
6.2.7 User Variables.....	12
6.3 Virtual Machine Execution Features.....	13
6.4 Arithmetic	13
6.5 Exception Handling	14
6.6 Resources	15
6.7 Programs and Tokens.....	15
7 System Services	16
7.1 Time Handling.....	16
7.2 Devices and I/O Services	17
7.3 Database Services.....	17
7.3.1 The Database Parameter Block.....	19
7.3.2 Database Instantiation	21
7.3.3 Database Exception Handling.....	22
7.4 Language and Message Handling	22
7.5 TLV Services	23
7.5.1 Basic Principles.....	23
7.5.2 TLV Definitions	23
7.5.3 TLV References	24
7.6 Hot Card List Management.....	25
7.7 Cryptographic Services	25
7.7.1 Modulo Multiplication.....	26
7.7.2 Secure Hash Algorithm (SHA-1)	26
7.7.3 Modulo Exponentiation.....	27
7.7.4 Long Shift.....	27
7.7.5 Long Subtract	27
7.7.6 Incremental Secure Hash Algorithm (SHA-1)	27
7.7.7 Cyclic Redundancy Check (CRC)	28
7.7.8 DES Key Schedule.....	28
7.7.9 DES encryption/decryption	28

7.8	Vectored Execution Sockets	28
7.8.1	CSS Functions	29
7.8.2	Socket Security	29
7.8.3	Socket Organisation	29
7.9	Module Handling Services	29
7.9.1	Module Loading by MODEXECUTE	30
7.9.2	Module Loading Procedure	32
7.9.3	Module Loading by MODCARDEXECUTE	35
8	Token Set Definition	37
8.1	Overview	37
8.2	Conventions	37
8.2.1	Number Formats	37
8.2.2	Token Descriptions	38
8.2.3	Branch and Code Offsets	38
8.2.4	Addresses	38
8.3	Data Typing	39
8.4	Token Compression	39
8.4.1	Optimised Data Access	39
8.4.2	Special Procedure Calls	39
8.4.3	Quoting	39
8.5	Prefix Tokens	40
8.6	Stack Manipulation Tokens	41
8.7	Data Access Tokens	43
8.8	Literal Tokens	45
8.9	Address Generation Tokens	46
8.10	Arithmetic Tokens	47
8.11	Relational Tokens	51
8.12	String Tokens	53
8.13	Frame Tokens	56
8.14	Extensible Memory Tokens	58
8.15	Flow of Control Tokens	59
8.15.1	Branch Tokens	59
8.15.2	Call Tokens	60
8.15.3	Loop Tokens	61
8.15.4	Hybrid Tokens	62
8.15.5	Quoting Tokens	63
8.16	Exception Tokens	63
8.17	Date, Time, and Timing Tokens	64
8.18	Generic Device I/O Tokens	64
8.19	Formatted I/O Tokens	68
8.20	Integrated Circuit Card Tokens	69
8.21	Magnetic Stripe Tokens	70
8.22	Socket Tokens	71
8.23	Database Services Tokens	72
8.24	Language and Message Tokens	77
8.25	TLV Tokens	78
8.25.1	TLV Buffer Access	78
8.25.2	TLV Processing	80
8.25.3	TLV Sequence Access	81
8.26	Hot Card List Tokens	82
8.27	Cryptographic Algorithm Token	83
8.28	Module Management Tokens	83
8.29	Operating System Interface Tokens	84
8.30	Miscellaneous Tokens	84
9	Module Delivery Format	85
9.1	Module ID Format	86
9.2	Socket List	86
9.3	Relocation Section	87
9.4	Module Import List	88

9.5	Module Export List	88
9.6	Module Procedure List.....	89
Annex A	(normative) OTA Token Lists	90
Annex B	(normative) Exceptions and I/O Return Codes	97
Annex C	(normative) Device Control	101
Annex D	(normative) Operating System Calls	116
Annex E	(normative) Rules for Using a Data Object List (DOL).....	117
Annex F	(informative) System Overview	118
Bibliography	137

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 20060 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 17, *Cards and personal identification*.

This second edition cancels and replaces the first edition (ISO/IEC 20060:2001), which has been technically revised.

Introduction

This International Standard specifies the Open Terminal Architecture (OTA) consistent with requirements and capabilities defined by documents [1] thru [8] in the Bibliography.

The overall architecture of the OTA is described in Annex F and is based on a virtual machine (VM) that can be programmed using high-level languages such as Forth or C. For compactness and efficiency, a tokenised form has been developed for delivering compiled programs to terminals of all CPU types. This and other virtual machine related issues are explained in Clause 5.

This International Standard defines a set of functions to be implemented in terminals in terms of instructions for a virtual machine. With these functions the application programmer is able to generate application software that is compact, portable and certifiable on all OTA terminals.

The inclusion of a function is determined by three main criteria:

- core compactness,
- execution speed,
- security requirements.

In this International Standard, the word “shall” indicates mandatory behaviour. The word “will” indicates predicted or consequential behaviour. The word “may” indicates permitted behaviour. The phrase “may not” indicates prohibited behaviour.

Information technology — Open Terminal Architecture (OTA) — Virtual machine

1 Scope

This International Standard provides the specifications for the standard Open Terminal Architecture (OTA) kernel in several layers:

- definition of the virtual machine (VM);
- description of the services provided by the VM to terminal programmers;
- specification of a set of tokens representing the native machine language of the VM;
- specification of the format in which token modules are delivered to an OTA kernel for processing.

OTA defines a standard software kernel whose functions and programming interface are common across all terminal types. This kernel is based on a standard “virtual machine,” which is implemented on each CPU type and which provides drivers for the terminal's I/O and all low-level CPU-specific logical and arithmetic functions. High-level libraries, terminal programs and payment applications may be developed using these standard kernel functions.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

None.