INTERNATIONAL STANDARD

ISO 11783-13

Second edition
2011-04-01

Tractors and machinery for agriculture and forestry — Serial control and communications data network —

Part 13: File server

Tracteurs et matériels agricoles et forestiers — Réseaux de commande et de communication de données en série —
Partie 13: Serveur de fichiers

Reference number
ISO 11783-13:2011(E)

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

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

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

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

ISO 11783-13 was prepared by Technical Committee ISO/TC 23, *Tractors and machinery for agriculture and forestry*, Subcommittee SC 19, *Agricultural electronics*.

This second edition cancels and replaces the first edition (ISO 11783-13:2007), of which it constitutes a minor revision.

ISO 11783 consists of the following parts, under the general title *Tractors and machinery for agriculture and forestry — Serial control and communications data network*:

— *Part 1: General standard for mobile data communication*

— *Part 2: Physical layer*

— *Part 3: Data link layer*

— *Part 4: Network layer*

— *Part 5: Network management*

— *Part 6: Virtual terminal*

— *Part 7: Implement messages application layer*

— *Part 8: Power train messages*

— *Part 9: Tractor ECU*

— *Part 10: Task controller and management information system data interchange*

— *Part 11: Mobile data element dictionary*

— *Part 12: Diagnostics services*

— *Part 13: File server*

— *Part 14: Sequence control*

Introduction

Parts 1 to 14 of ISO 11783 specify a communications system for agricultural equipment based on ISO 11898-1[1] and ISO 11898-2[2]. SAE J1939[3] documents, on which parts of ISO 11783 are based, were developed jointly for use in truck and bus applications and for construction and agriculture applications. Joint documents were completed to allow electronic units that meet the truck and bus SAE J1939 specifications to be used by agricultural and forestry equipment with minimal changes. General information on ISO 11783 is to be found in ISO 11783-1.

The purpose of ISO 11783 is to provide an open, interconnected system for on-board electronic systems. It is intended to enable electronic control units (ECUs) to communicate with each other, providing a standardized system.

The International Organization for Standardization (ISO) draws attention to the fact that it is claimed that compliance with this part of ISO 11783 may involve the use of a patent concerning the controller area network (CAN) protocol referred to throughout the document.

ISO takes no position concerning the evidence, validity and scope of this patent.

The holder of this patent has assured ISO that he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO. Information may be obtained from:

Robert Bosch GmbH
Wernerstrasse 51
Postfach 30 02 20
D-70442 Stuttgart-Feuerbach
Germany

Attention is drawn to the possibility that some of the elements of this part of ISO 11783 may be the subject of patent rights other than those identified above. ISO shall not be held responsible for identifying any or all such patent rights.

Tractors and machinery for agriculture and forestry — Serial control and communications data network —

Part 13: File server

1 Scope

ISO 11783 as a whole specifies a serial data network for control and communications on forestry or agricultural tractors and mounted, semi-mounted, towed or self-propelled implements. Its purpose is to standardize the method and format of transfer of data between sensors, actuators, control elements and information storage and display units, whether mounted on, or part of, the tractor or implement. This part of ISO 11783 specifies the file server (FS) for use by a tractor or self-propelled implement.

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.

ISO 11783-1, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 1: General standard for mobile data communication*

ISO 11783-3, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 3: Data link layer*

ISO 11783-5, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 5: Network management*

ISO 11783-6, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 6: Virtual terminal*

ISO 11783-7, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 7: Implement messages application layer*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 11783-1 and the following apply.

3.1
client
electronic control unit (ECU) on the mobile implement bus that uses the services of the file server

3.2
directory
file which stores administrative information about other files

3.3
file
data object that stores data on a storage device

3.4
file attribute
bit-coded information that defines the type and features of a file

3.5
file server
FS
electronic control unit (ECU) on the mobile implement bus that provides storage for files and uses a set of commands for the handling of, and access to, these files

3.6
filename
name conforming to requirements of a character set, which identifies a file or directory

NOTE See Annex A for the character set.

3.7
Handle
data object used for accessing files and directories

3.8
hidden attribute
file attribute indicating that the file should not appear in a directory listing

NOTE A client sets this attribute by using the file server (FS) commands.

3.9
path
specification of a filename that may also include the directory name

3.10
read-only attribute
file attribute used to prevent writing to, or deletion of, a file

NOTE A client sets this attribute by using the file server (FS) commands.

3.11
volume
directory that refers to a specific logical or physical storage unit or space

NOTE The primary volume is the volume used as current volume when the file server (FS) is started.

4 General

The message set specified in this part of ISO 11783 is designed to support the needs of tractors and implements in using the services of a file server (FS) — a distinct electronic control unit (ECU) on the mobile implement control system that enables all controllers to store or retrieve data from a file-based storage device.

5 Requirements

5.1 General message format

The general message format uses the parameter group number as the label for a group of parameters (see Annex B). Each of the parameters within the group can be expressed as characters, as scaled data defined by the ranges given in 5.2, or as one or more bits. Characters shall be transmitted with the left-most character first. Numerical parameters consisting of two or more data bytes shall be transmitted least significant byte (LSB) first. When variable-length messages have eight or less data bytes, these messages shall be transmitted in a single controller area network (CAN) frame. When variable length messages have nine or more data bytes, the transport protocol (TP), in accordance with ISO 11783-3, or the extended transport protocol (ETP), in accordance with ISO 11783-6, is required. When a message has less than eight data bytes, the unused bytes shall be filled with FF_{16} values.

5.2 File data format

5.2.1 Data

Data consists of a block of bytes (unsigned eight-bit values). All values in the range of 0_{10} … 255_{10}, 00_{16} … FF_{16} are allowed. There is no special handling of individual characters (control characters, end-of-line markers, end-of-file markers or similar characters).

5.2.2 Bit groups

Groups of one to eight bits are packed into one byte as bit 7 … bit 0. Groups of nine to 16 bits are packed into two bytes in the order of LSB as bit 7 … bit 0, followed by most significant byte (MSB) as bit 15 … bit 8. Unused bits in a bit group default to a value of 0 (zero).

5.2.3 Integer

Unsigned 8 bit	1 byte	$0 … 2^8–1$	0_{10} … 255_{10}
Unsigned 16 bit	2 bytes, LSB first	$0 … 2^{16}–1$	0_{10} … 65535_{10}
Unsigned 32 bit	4 bytes, LSB first	$0 … 2^{32}–1$	0_{10} … 4294967295_{10}
Signed 32 bit	4 bytes, LSB first, two's compliment	$–2^{31} … 2^{31}–1$	$–2147483648_{10}$ … $+2147483647_{10}$

5.2.4 Character string

A string contains characters represented by bytes (unsigned eight-bit values). The length of a string is specified by a string length data item. Annex A specifies the characters allowed in a string used as a filename or a path name.

5.3 Data transmission control

5.3.1 General

Each communication transaction between a client and the FS is initiated by a request from the client and terminated by a response from the FS. In order to provide fail-safe communications, it is important that the client *assign* the received response to a corresponding request and repeat an erroneous request without triggering the complete execution again.

5.3.2 Strategy

The client can issue a request and receive no response because of transient communication problems. The failure can happen during the request message, i.e. the FS does not receive the request, or during the response message, i.e. the client does not receive the response. The client cannot distinguish between these two cases and shall repeat the request to obtain the requested data.

If there is no transaction strategy, the problem of the FS not receiving the request is resolved by the client sending a second request and the FS responding with the requested data. However, if the client does not receive the correct response data message and sends a second request, the FS then sends the next data from the file; this is because a data request automatically advances to the next data in the file.

A transaction strategy is therefore required to prevent such errors. Each client on the network maintains its own transaction number (TAN) counter, which should start at 0 after a power cycle.

Each client generates a TAN for each request that it sends to the FS. This is done by incrementing the last TAN used for the next request. The client is responsible for checking that a received response contains the same TAN that was used in the request during the communication session, thus ensuring that there are no lost commands. The FS shall remember the last command processed and response message sent for each client. This is done by incrementing the last TAN for the next request. The FS compares each new request with the previous request from the same client. If the TAN is not the same, the request is implemented and the response is sent. If the TAN is the same as the previously received request, the request is not implemented and the previous response is sent. Thus if the client sends a second request, in the case where the FS never received the first request, the FS *receives* the TAN for the first time, *implements* the request and sends the correct data response. If the FS receives a request with the same TAN that it has already received, it does not implement the request, but the previous response is retransmitted.

5.3.3 Timeout

The execution time of all FS commands (the time between request and response) is maintained within reasonable limits. The client shall monitor the time while waiting for a response.

The timeouts specified in ISO 11783-3 for the transport protocol and in ISO 11783-6 for the extended transport protocol shall be met for the execution of commands.

If a timeout expires, the request is assumed to have failed and the client shall repeat the request using the same TAN.

If a request response takes longer than 200 ms after the completion of the request, the FS shall send the status message to indicate busy state to the client. This provides a request timeout of 600 ms if the FS status message does not show a busy status.

5.4 Date and time support

Several FS commands require a file date and time. UTC[1)] is used for this time. The file server's implementation of real time support can be either by maintaining its own real time information or by requesting the time and date information using the Time/Date parameter group specified in ISO 11783-7. The date and time of a file is the latest date and time when a file was actually modified. A file which is opened for read/write access but is not modified by a write action shall not get a later date and time.

5.5 Multi-client support

The file server shall support one or more clients. If more than one client has a connection simultaneously, the FS shall function with each client as if it is the only one on the network. There shall be no interference between the commands processed for different clients.

---

1) Coordinated universal time, or universal time, formerly known as Greenwich mean time (GMT).

Upon connection of a client, the file server initiates the current directory for that client as the root directory of the primary volume of the FS file system. If there are no volumes, then the current directory is set to the list of volumes "\\". The client is required to use the appropriate Change Current Directory or Open File commands to access files that need to be unique for that client. In the case where multiple clients require access to common files, these clients are responsible for synchronizing their directory and file naming conventions to enable access to these common files. To prevent unintentional access to manufacturer proprietary files, a reserved directory name containing the manufacturer code according to ISO 11783-5 is specified. The naming convention of the manufacturer-specific directory is:

MCMC0000

where 0000 contains the four-digit manufacturer code in decimal representation, formatted with leading zeroes. A client shall not use this manufacturer-coded directory name using a manufacturer code other than the manufacturer code in its NAME field. When the client attempts to open a file in a manufacturer-specific directory where the manufacturer code in the NAME of that client is not that of the manufacturer-specific directory name, the FS shall prevent access and return an "access denied" error code.

When a file server supports multiple volumes, manufacturer-specific directories can be created on each volume. Creation of a manufacturer-specific directory is the responsibility of the client. The manufacturer-specific directory shall only be placed at the root of each volume.

5.6 File Handles

An FS may support multiple file Handles. Many of the commands available for the FS create and/or use file Handles. However, there are some commands that only use folder or filenames. Internally, if the FS creates a file Handle to process these commands, the number of open files shall be incremented to reflect the internal status of the FS.

5.7 Volumes

Different types of media (FLASH, removable media, ruggedized disk drives) can be assigned to different volumes.

An FS can support multiple volumes. It is also possible for an FS to list no volumes — for example, with uninitialized media or no device found.

The list of volumes, specified by "\\", is the highest layer (or base) of a directory structure.

A special service tool for the FS can be used to create volumes as specified in Annex C, Initialize Volume Request (C.5.2.2) message. The names of the volumes are determined by the FS; however, the FS can allow a service tool to name them as specified for this message.

NOTE This part of ISO 11783 does not specify *how* the service tool selects the media or volumes to initialize if they are not named and listed in the list of volumes "\\".

Annex A
(normative)

Character set

A.1 Valid characters

The file server uses filenames and path names. Every character used for one of the 8.3 names (see below) is validated by the FS using the appropriate subset of Table A.1 (based on ISO/IEC 8859-1) and the filename and path definitions given in A.2. Only printable characters are visible when presenting the filename or path name to a user. For case-insensitive file systems, the FS shall convert the lower-case characters (61_{16} to $7A_{16}$) to upper-case (41_{16} to $5A_{16}$). File servers that do not support long filenames shall use an 8.3 name and extension notation where the name is a maximum of eight characters long, optionally followed by an extension that starts with a single dot ($2E_{16}$), and finishes with a maximum of three characters. File servers that support long filenames shall use the filename and path defined in A.2.

Table A.1 — ISO Latin 1 character set

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂
1	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂
2	space	!	“	#	$	%	&	‘	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	✂
8	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂
9	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂	✂
A	✂	¡	¢	£	¤	¥	¦	§	¨	©	«	«	¬	-	®	¯
B	°	±	²	³	³	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

✂ non-printable character

NOTE This information is given for reference; the following does not use all the available characters given here.

A.2 Filename and path definitions

A.2.1 General

Definitions:

[ ] any of the characters in Table A.1, including none from the set (optional);

[A-B] defines an inclusive range from the first through the last;

( ) group;

< > character class;

\ escapes the following character, as in "\[", which indicates a single left bracket, not the containment of a set;

A | B sequence "A" or "B";

A + B sequence of A followed by B;

{m} exactly m of the preceding set;

{m,n} from m up to and including n of the preceding set;

\xXX character code in hexadecimal notation where XX are two hexadecimal digits (\x20, for example, indicates character code 32, which is the space character).

A.2.2 Name definitions

A.2.2.1 Names

Names are from one to 254 characters in length, using the character set given below. The commonly used file systems given in Annex D were used to determine name restrictions that would allow these names to be used with minimal feature loss.

ShortNameChar ::= [0-9A-Z\-!#$%&'\(\)@^_`\{\}~\xA1-\xFF] {1}

LongNameChar ::= any single character defined by Unicode/ISO/IEC 10646 except NUL, \, *, or ?

WildCardChar ::= [*?] {1}

ManufacturerSpecificDirectoryChar ::= [~] {1}

PathSeparatorChar ::= [\\ ] {1}

VolumeListIndicator ::= [\\ ] {2}

ParentFolderIndicator ::= [.] {2}

CurrentFolderIndicator ::= [.] {1}

MfgSpecificFolderIndicator ::= [<ManufacturerSpecificDirectoryChar>] {1}

ShortWildCardNameChar::= [<ShortNameChar> | <WildCardChar>] {1}

LongWildCardNameChar::= [<LongNameChar> | <WildCardChar>] {1}

ShortName ::= [<ShortNameChar>] {1,8} [.+ [<ShortNameChar>] {0,3}]

LongName ::= [<LongNameChar>] {1,254}

ShortWildCardName ::= [<ShortWildCardNameChar>] {1,8} [.+ [<ShortWildCardNameChar>] {0,3}]

LongWildCardName ::= [<LongWildCardNameChar>] {1,254}

A.2.2.2 Filenames

Filenames use the names defined in A.2.2.1.

ShortFileName ::= <ShortName>

LongFileName ::= <LongName>

EXAMPLE Test, Test.txt, Test Filename.long.name (specifically a LongName).

A.2.2.3 Volumes

Volumes use the names defined in A.2.2.1.

VolumeName ::= <LongName>

EXAMPLE VOL_B, Flash Volume (specifically a LongName).

A.2.3 Path definitions

A.2.3.1 General

A path definition is similar to a filename definition but has additional prefix definitions and delimiters between path segments.

When a directory listing from path “\\” (two backslashes) is requested, the FS shall return a list of volumes. All file servers shall support “\\” so that the clients can query the volumes (including removable media), even if the FS has only one volume.

The two predefined special directory names, “.” and “..”, refer to the current (“.”) and parent (“..”) directories. These predefined directory names shall not be reported in a directory listing but may be used in a path name to specify reference to a current or parent directory.

The “~” character (*tilde*) may be used as a placeholder for the manufacturer-specific directory of a client. This character can only be specified at the beginning of a path or after a volume name and shall be replaced by the FS with the manufacturer-specific directory name on the current volume. If there is no current volume, then the server primary volume shall be used. The “~” can be used in the name, but cannot be the only character, since this would be interpreted as the manufacturer-specific directory. For example, “~\file1.txt” to “MCMC0000\file1.txt”.

ShortFolderName ::= [<ShortName> | <ParentFolderIndicator> | <CurrentFolderIndicator>] {1}

LongFolderName ::= [<LongName> | <ParentFolderIndicator> | <CurrentFolderIndicator>] {1}

ShortPathName ::= [

[<VolumeListIndictor>] |

[[<VolumeListIndictor>] + <VolumeName> + <PathSeparatorChar> + [<MfgSpecificFolderIndicator> + <PathSeparatorChar>] {0,1} + [<ShortFolderName> + <PathSeparatorChar>] {0,n}] |

[[<PathSeparatorChar>] {0,1} + [<ShortFolderName> + <PathSeparatorChar>] {0,n}] |

[[<MfgSpecificFolderIndicator> + <PathSeparatorChar>] {0,1} + [<ShortFolderName> + <PathSeparatorChar>] {0,n}]

] {1}

LongPathName ::= [

[<VolumeListIndictor>] |

[[<VolumeListIndictor>] + <VolumeName> + <PathSeparatorChar> + [<MfgSpecificFolderIndicator> + <PathSeparatorChar>] {0,1} + [<LongFolderName> + <PathSeparatorChar>] {0,n}] |

[[<PathSeparatorChar>] {0,1} + [<LongFolderName> + <PathSeparatorChar>] {0,n}] |

[[<MfgSpecificFolderIndicator> + <PathSeparatorChar>] {0,1} + [<LongFolderName> + <PathSeparatorChar>] {0,n}]

] {1}

EXAMPLE 1 Path relative to current directory:

.\

..\path\

..\Long path name\ (specifically a LongName)

Path\

Level1\Level2\

Path.dir\

Long path name\ (specifically a LongName)

EXAMPLE 2 Path relative to root of current volume:

\Path\

\Level1\Level2\

\Path.dir\

\Long path name\ (specifically a LongName)