SLOVENSKI STANDARD
oSIST prEN IEC 62541-20:2024

01-marec-2024

Enotna arhitektura OPC - 20. del: Prenos datotek

OPC unified architecture - Part 20: File transfer

Ta slovenski standard je istoveten z: **prEN IEC 62541-20:2024**

ICS:

25.040.40	Merjenje in krmiljenje industrijskih postopkov	Industrial process measurement and control
35.240.50	Uporabniške rešitve IT v industriji	IT applications in industry

oSIST prEN IEC 62541-20:2024 **en,fr,de**

65E/1045/CDV

COMMITTEE DRAFT FOR VOTE (CDV)

PROJECT NUMBER: **IEC 62541-20 ED1**	
DATE OF CIRCULATION: **2024-01-26**	CLOSING DATE FOR VOTING: **2024-04-19**
SUPERSEDES DOCUMENTS: **65E/955/NP, 65E/1015/RVN**	

IEC SC 65E : DEVICES AND INTEGRATION IN ENTERPRISE SYSTEMS

SECRETARIAT: United States of America	SECRETARY: Mr Donald (Bob) Lattimer
OF INTEREST TO THE FOLLOWING COMMITTEES:	PROPOSED HORIZONTAL STANDARD: □ Other TC/SCs are requested to indicate their interest, if any, in this CDV to the secretary.

FUNCTIONS CONCERNED:

□ EMC □ ENVIRONMENT □ QUALITY ASSURANCE □ SAFETY

☒ SUBMITTED FOR CENELEC PARALLEL VOTING **Attention IEC-CENELEC parallel voting** The attention of IEC National Committees, members of CENELEC, is drawn to the fact that this Committee Draft for Vote (CDV) is submitted for parallel voting. The CENELEC members are invited to vote through the CENELEC online voting system.	□ NOT SUBMITTED FOR CENELEC PARALLEL VOTING

This document is still under study and subject to change. It should not be used for reference purposes.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Recipients of this document are invited to submit, with their comments, notification of any relevant "In Some Countries" clauses to be included should this proposal proceed. Recipients are reminded that the CDV stage is the final stage for submitting ISC clauses. (SEE AC/22/2007 OR NEW GUIDANCE DOC).

TITLE:

OPC Unified Architecture – Part 20: File Transfer

PROPOSED STABILITY DATE: 2026

NOTE FROM TC/SC OFFICERS:

CONTENTS

FIGURES

TABLES

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

OPC UNIFIED ARCHITECTURE –

Part 20: File Transfer

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as “IEC Publication(s)”). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

The main task of IEC technical committees is to prepare International Standards. However, a technical committee may propose the publication of a technical report when it has collected data of a different kind from that which is normally published as an International Standard, for example "state of the art".

International Standard IEC 62541-20 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

The text of this international standard is based on the following documents:

CDV	Report on voting
65E/XX/CDV	65E/XX/RVC

Full information on the voting for the approval of this international standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

Throughout this document and the other Parts of the series, certain document conventions are used:

Italics are used to denote a defined term or definition that appears in the "Terms and definition" clause in one of the parts of the series.

Italics are also used to denote the name of a service input or output parameter or the name of a structure or element of a structure that are usually defined in tables.

The *italicized terms* and *names* are also often written in camel-case (the practice of writing compound words or phrases in which the elements are joined without spaces, with each element's initial letter capitalized within the compound). For example, the defined term is *AddressSpace* instead of Address Space. This makes it easier to understand that there is a single definition for *AddressSpace*, not separate definitions for Address and Space.

A list of all parts of the IEC 62541 series is included in IEC 62541-1 clause 4 Structure of the OPC UA series and published under the general title OPC Unified Architecture, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

OPC Unified Architecture Specification

Part 20: File Transfer

1 Scope

This part of the OPC Unified Architecture defines an Information Model. The Information Model describes the basic infrastructure to model file transfers.

Note: In the previous version, File Transfer was in IEC 62541-5, Annex C

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments and errata) applies.

IEC 62541-1, *OPC Unified Architecture – Part 1: Overview and Concepts*

IEC 62541-3, *OPC Unified Architecture – Part 3: Address Space Model*

IEC 62541-4, *OPC Unified Architecture – Part 4: Services*

IEC 62541-5, *OPC Unified Architecture – Part 5: Information Model*

3 Terms, definitions, abbreviated terms, and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62541-1, IEC 62541-3 and IEC 62541-5 apply.

4 File Transfer Model

4.1 Overview

This document describes an information model for file transfer. Files could be modelled in OPC UA as simple Variables using ByteStrings. However, the overall message size in OPC UA is limited due to resources and security issues (denial of service attacks). Only accessing parts of the array can lead to concurrency issues if one client is reading the array while others are manipulating it. Therefore, the *ObjectType FileType* is defined representing a file with *Methods* to access the file. The life-cycle of a file stored on a hard disk and an instance of the *FileType* representing the file in an OPC UA *AddressSpace* can be independent.

In addition to representing individual files this document also defines a way to represent a whole file system or a part of a file system. This can be done using the *FileDirectoryType* in combination with the *FileType*. The *FileDirectoryType* provides *Methods* to create, delete, and move files and directories. The root of a file system or part of a file system is represented by an instance of the *FileDirectoryType* with the *BrowseName FileSystem*. All directories below the root directory are represented by instances of the *FileDirectoryType* or a subtype. All files below the root directory are represented by instances of the *FileType* or a subtype.

In different situations like transfer of configuration files or firmware update, the files are temporary, and an additional handshake is necessary to create the file for reading or to apply the file after writing it to the server. This use case is covered by the *TemporaryFileTransferType* defined in this document.

4.2 FileType

4.2.1 General

This *ObjectType* defines a type for files. It is formally defined in Table 1.

Table 1 – FileType

Attribute	**Value**				
BrowseName	FileType				
IsAbstract	False				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Subtype of the BaseObjectType defined in IEC 62541-5					
HasProperty	Variable	Size	UInt64	PropertyType	Mandatory
HasProperty	Variable	Writable	Boolean	PropertyType	Mandatory
HasProperty	Variable	UserWritable	Boolean	PropertyType	Mandatory
HasProperty	Variable	OpenCount	UInt16	PropertyType	Mandatory
HasProperty	Variable	MimeType	String	PropertyType	Optional
HasComponent	Method	Open	Defined in 4.2.2		Mandatory
HasComponent	Method	Close	Defined in 4.2.3		Mandatory
HasComponent	Method	Read	Defined in 4.2.4		Mandatory
HasComponent	Method	Write	Defined in 4.2.5		Mandatory
HasComponent	Method	GetPosition	Defined in 4.2.6		Mandatory
HasComponent	Method	SetPosition	Defined in 4.2.7		Mandatory
HasProperty	Variable	MaxByteStringLength	UInt32	PropertyType	Optional
HasProperty	Variable	LastModifiedTime	DateTime	PropertyType	Optional
Conformance Units					
Base Info FileType Base					

Size defines the size of the file in Bytes. When a file is opened for write, the size might not be accurate. If the *Server* can not accurately determine the size of the file, the Size *Property* shall be returned to a *Client* with a *StatusCode* of *Bad_NotSupported*.

Writable indicates whether the file is writable. It does not take any user access rights into account, i.e., although the file is writable this may be restricted to a certain user / user group. The *Property* does not consider whether the file is currently opened for writing by another client and thus currently locked and not writable by others.

UserWritable indicates whether the file is writable taking user access rights into account. The Property does not consider whether the file is currently opened for writing by another client and thus currently locked and not writable by others.

OpenCount indicates the number of currently valid file handles on the file.

The optional *Property MimeType* contains the media type of the file based on RFC 2046.

Note that all *Methods* on a file require a fileHandle, which is returned in the *Open Method*.

The optional *MaxByteStringLength Property* indicates the maximum number of bytes of the read and write buffers. If this *Property* is not present then the maximum size is defined by the *MaxByteStringLength Property* of the *ServerCapabilitiesType* defined in IEC 62541-5.

The optional *LastModifiedTime Property* indicates the time the file was last modified. The *Property* shall be updated whenever the *Server* detects that the file has changed.

4.2.2 Open

Open is used to open a file represented by an *Object* of FileType. When a client opens a file it gets a file handle that is valid while the session is open. Clients shall use the Close *Method* to release the handle when they do not need access to the file anymore. Clients can open the same file several times for read. A request to open for writing shall return Bad_NotWritable when the file is already opened. A request to open for reading shall return Bad_NotReadable when the file is already opened for writing.

Signature

```
  Open(
     [in] Byte mode
     [out] UInt32 fileHandle
    );
```

<table>
<tr><th>Argument</th><th>Description</th></tr>
<tr><td>mode</td><td>Indicates whether the file should be opened only for read operations or for read and write operations and where the initial position is set.
The <i>mode</i> is an 8-bit unsigned integer used as bit mask with the structure defined in the following table:
<table>
<tr><th>Field</th><th>Bit</th><th>Description</th></tr>
<tr><td>Read</td><td>0</td><td>The file is opened for reading. If this bit is not set the Read Method cannot be executed.</td></tr>
<tr><td>Write</td><td>1</td><td>The file is opened for writing. If this bit is not set the Write Method cannot be executed.</td></tr>
<tr><td>EraseExisting</td><td>2</td><td>This bit can only be set if the file is opened for writing (Write bit is set). The existing content of the file is erased and an empty file is provided.</td></tr>
<tr><td>Append</td><td>3</td><td>When the Append bit is set the file is opened at end of the file, otherwise at begin of the file. The SetPosition Method can be used to change the position.</td></tr>
<tr><td>Reserved</td><td>4:7</td><td>Reserved for future use. Shall always be zero.</td></tr>
</table></td></tr>
<tr><td>fileHandle</td><td>A handle for the file used in other method calls indicating not the file (this is done by the Object of the Method call) but the access request and thus the position in the file. The fileHandle is generated by the server and is unique for the Session. Clients cannot transfer the fileHandle to another Session but need to get a new fileHandle by calling the Open Method.</td></tr>
</table>

Method Result Codes (defined in Call Service)

Result Code	Description
Bad_NotReadable	See IEC 62541-4 for a general description. File might be locked and thus not readable.
Bad_NotWritable	See IEC 62541-4 for a general description.
Bad_InvalidState	See IEC 62541-4 for a general description. The file is locked and thus not writable.
Bad_InvalidArgument	See IEC 62541-4 for a general description. Mode setting is invalid.
Bad_NotFound	See IEC 62541-4 for a general description.
Bad_UnexpectedError	See IEC 62541-4 for a general description.